**Universidade de Aveiro**
**2010**
Departamento de Electrónica, Telecomunicações e Informática

**Hélder Troca Zagalo**

# Arquitectura e Plataforma de *Middleware* para Suporte de Bibliotecas Digitais Distribuídas e Ecléticas

**Universidade de Aveiro**
**2010**

Departamento de Electrónica, Telecomunicações e Informática

**Hélder Troca Zagalo**

# Arquitectura e Plataforma de *Middleware* para Suporte de Bibliotecas Digitais Distribuídas e Ecléticas

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Informática, realizada sob a orientação científica do Doutor Joaquim Arnaldo Martins, Professor Catedrático do Departamento de Engenharia Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Joaquim Sousa Pinto, Professor Auxiliar do Departamento de Engenharia Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

Ao meu filho.

**o júri**

presidente — Reitor da Universidade de Aveiro

Prof. Doutor Joaquim Arnaldo Carvalho Martins
Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Prof. Doutor José Luís Brinquete Borbinha
Professor Auxiliar do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Doutora Ana Alice Rodrigues Pereira Baptista
Professora Auxiliar do Departamento de Sistemas de Informação da Escola de Engenharia da Universidade do Minho

Prof. Doutor Joaquim Manuel Henriques de Sousa Pinto
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Prof. Doutor José Manuel Matos Moreira
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

**agradecimentos**

Agradeço ao Prof. Joaquim Arnaldo Martins, o principal responsável pela supervisão do trabalho de investigação e desenvolvimento, conducente à presente dissertação. A sua grande competência científica e os seus amplos conhecimentos em áreas diversas da Informática, permitiram uma orientação de proximidade, geradora de inúmeras discussões, que levaram a um grande aprimoramento das ideias e à sua mais fácil concretização. Expresso ainda o meu apresso pelas suas qualidades humanas, ao manter sempre um elevado sentido de suporte moral, principalmente nos momentos de maior dificuldade por que passou este trabalho.

Agradeço também ao Prof. Joaquim Sousa Pinto, pela sua orientação e crítica do trabalho, que em muito ajudaram à sua prossecução. Também o seu suporte moral, foi muitas vezes decisivo para vencer barreiras que eu achava intransponíveis.

Ao Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA) pelos meios que colocou à minha disposição para levar a cabo o trabalho de investigação.

À Universidade de Aveiro pelo apoio financeiro, através de uma bolsa de doutoramento com a duração de um ano.

À Fundação para Ciência e Tecnologia (FCT), pelo seu apoio financeiro através de uma bolsa de doutoramento (PRAXIS XXI – BD/21361/99), que usufruí entre Janeiro de 2000 e Setembro de 2001.

Um profundo agradecimento aos meus pais que foram desde sempre a pedra basilar que possibilitou todos os esforços conducentes a todas as minhas formações, nas quais se inclui de forma preeminente este projecto de doutoramento.

Um agradecimento muito elevado, pleno de amizade e sentimento, ao meu irmão Nelson Zagalo, pelo seu suporte moral, empenhamento e ajuda no término deste projecto.

Por fim, uma palavra especial de apreço a todas as pessoas amigas pelo seu encorajamento.

**palavras-chave**

bibliotecas digitais, sistemas de informação, sistemas distribuídos, *middleware*, interoperabilidade, metadados

**resumo**

O trabalho apresentado nesta dissertação teve por objectivo principal a concepção, modelação e desenvolvimento de uma plataforma de *middleware* que permitisse a integração de sistemas de informação, em todos os seus níveis (dados, lógico e apresentação), perfazendo uma federação de bibliotecas digitais distribuídas e ecléticas.

Para este fim, foram estudadas as várias abordagens de modelação e organização das bibliotecas digitais, assim como os diversos sistemas e tecnologias de suporte existentes no momento inicial do trabalho.

Compreendendo a existência de muitas lacunas ainda neste domínio, nomeadamente ao nível da interoperabilidade de sistemas heterogéneos e integração da semântica de metadados, decidiu-se proceder a um trabalho de investigação e desenvolvimento que pudesse apresentar eventuais soluções para o preenchimento de tais lacunas.

Desta forma, surgem neste trabalho duas tecnologias, o XML e o Dublin Core, que servem de base a todas as restantes tecnologias usadas para a interoperabilidade e para a integração.

Ainda utilizando estas tecnologias base, foram estudados e desenvolvidos meios simples, mas eficientes, de salvaguarda, indexação e pesquisa de informação, tentando manter a independência face aos grandes produtores de bases de dados, que só por si não resolvem alguns dos problemas mais críticos da investigação no domínio das bibliotecas digitais.

**keywords**

digital libraries, information systems, distributed systems, middleware, interoperability, metadata

**abstract**

The main objective of the work presented in this dissertation is the design, modulation and development of a middleware framework to allow information systems interoperability, in all their scope (data, logic and presentation), to accomplish a distributed and eclectic digital libraries federation.
Several modulations and organizations were approached, and several support systems and technologies were studied.
Understanding the existence of many gaps in this domain, namely in heterogeneous information systems interoperation and metadata semantic integration, it was decided to conduct a research and development work, which, eventually, could present some solutions to fill in these gaps.
In this way, two technologies, XML and Dublin Core, appear to serve as the basis of all remaining technologies, to interoperate and to achieve semantic integration.
Using yet these technologies, it was also studied and developed simple means, but efficient ones, to save, index and query information, preserving the independence from major data base producers, which by their selves don't solve critical problems in the digital libraries research domain.

# Índice Geral

# Lista de Figuras

# Capítulo 1

# Introdução

## 1.1 O Problema

A lógica que orientou o aparecimento deste trabalho foi a da engenharia, que, como tal, exerceu a sua força não apenas no campo de uma necessidade aplicativa mas também objectivada pela resolução de problemas.

Assim, e do ponto de vista deste projecto de doutoramento, existe um problema específico relacionado com a necessidade de acesso a informação distribuída e residente em fontes heterogéneas, tanto ao nível dos sistemas como dos dados. Um problema que é fruto da evolução tecnológica, ao potenciar a criação de redes de comunicação, que permitiram a interconexão entre diferentes tipos de sistemas e diferentes tipos de dados e levaram a aumentos exponenciais do processamento de informação.

Esta evolução teve impactos no utilizador, tanto ao nível da velocidade como da quantidade de informação a que este pode e precisa de aceder. Deste modo, o impacto tecnológico criou uma clara necessidade, por parte da sociedade utilizadora, de um modo de aceder à informação, o mais holístico possível. Sabendo que a informação pode provir de fontes muito diferentes, é do interesse de quem acede, que essas diferenças sejam percepcionadas de forma a não perturbarem o trabalho de busca ou pesquisa do utilizador. Ou seja, interessa ao utilizador final, que a primeira perspectiva sobre a informação recolhida, para além de ser clara e legível, deve também ser independente

das características da sua proveniência: como as especificidades dos diferentes sistemas em que se encontra armazenada e dos diferentes formatos de dados que a compõem.

Sendo as bibliotecas digitais consideradas sistemas de informação por excelência, toda esta problemática não lhe é indiferente, sobretudo quando a tónica se coloca ao nível da interoperabilidade entre elas. É precisamente no contexto das bibliotecas digitais, que este projecto de doutoramento aborda o problema.

## 1.2 Os Objectivos

Com vista à proposição de uma solução para a problemática exposta antes, foi necessário estudar e avaliar o estado da arte dos acervos digitais, nomeadamente perceber de que modo a informação é tratada e acedida em diferentes sistemas, sabendo que o seus processos de criação têm sido bastante independentes e que o número de protocolos e normas de catalogação proliferam.

Após esse trabalho, estabeleceu-se como objectivos gerais, a proposta de uma solução de âmbito informático, capaz de oferecer:

- o aumento da interoperabilidade entre diferentes sistemas de repositórios digitais, utilizados na implementação de bibliotecas digitais distribuídas e heterogéneas, sabida que é, a necessidade da utilização de diferentes protocolos de pesquisa e de formatos de dados, aquando do seu acesso;
- a integração do acesso destes sistemas a partir de um ponto único, encontrando-se os sistemas distribuídos;
- a integração dos dados provindos das suas diversas fontes, promovendo uma visão mais holística da informação a manipular;
- meios alternativos para as funções de armazenamento e indexação em repositórios digitais, que possam ser mais adequados à pesquisa e recolha de dados que se encontram em formatos específicos.

Por forma a alcançar estes objectivos de ordem genérica, foram definidos os seguintes objectivos específicos:

- conceber uma arquitectura completa, desde o modelo funcional ao modelo de dados, para uma plataforma de *middleware* que permita o acesso integrado e paralelo a múltiplas bibliotecas digitais heterogéneas e distribuídas;
- conceber esta arquitectura por forma a que a plataforma seja uma plataforma de serviços, utilizando, para isso, a tecnologia aberta dos *web services*, promovendo a

homogeneidade no seio da heterogeneidade de protocolos de comunicação em uso;

- conceber uma interface funcional comum para todos os serviços a operarem na plataforma de *middleware*, assim como todo um conjunto de mensagens que circulam entre esses serviços;
- conceber a possibilidade de registo e descoberta destes serviços, de forma automatizada;
- conceber a integração e harmonização dos dados recebidos através da utilização do modelo de metadados Dublin Core, o que pressupõe a conversão entre diferentes modelos de metadados e a identificação e eliminação de metadados duplicados;
- conceber formas de utilizar o sistema de ficheiros como repositório digital, utilizando e desenvolvendo aplicativos específicos para garantir o armazenamento e a indexação de documentos no formato XML, visto este tipo particular de documento poder ser detentor de estruturas complexas, difíceis de serem totalmente processadas nas bases de dados tradicionais.

Na prossecução destes objectivos, deparou-se com alguns problemas principais de engenharia que foram redimidos no âmbito da investigação aplicada. São estes:

- a pesquisa distribuída por múltiplas e multifacetadas fontes de informação;
- o execução paralela, em tempo real, destas pesquisas, assim como da recepção dos resultados;
- a conversão entre diferentes modelos de metadados;
- a identificação de diferentes registos de metadados que descrevem um mesmo recurso;
- a identificação e recuperação de informação em elementos repetidos, no mesmo nível hierárquico de um documento XML, para a correcta indexação do mesmo.

A resolução destes problemas, na área particular das bibliotecas digitais, tornaram-se contributos no âmbito da investigação aplicada para esta área, contudo devem ainda ser consideradas as seguintes contribuições, neste âmbito e nesta área:

- a concepção de uma plataforma de *middleware* para a federação de repositórios e bibliotecas digitais, capaz de um elevado nível de escalabilidade;

- a concepção recursiva dos modelos funcional e de dados desta plataforma, o que permite uma concepção minimalista de um sistema de elevado grau de complexidade;
- e a utilização de uma tecnologia web emergente, à altura do início deste projecto, para a materialização de uma plataforma de *middleware* de serviços, promovendo a substituição de protocolos de índole mais específica por tecnologias de mais fácil e mais lata aplicação.

## 1.3 A Abordagem

Entre 1997 e 1999 surgiu em Portugal uma iniciativa, a iniciativa RUBI – Rede Universitária de Bibliotecas e Informação (RUBI, 1999), que tinha como objectivo prioritário a concepção e desenvolvimento de um "Catálogo Bibliográfico Distribuído". Este catálogo seria um catálogo virtual, sem existência física real, mas baseado e representando todos os catálogos reais pertencentes às bibliotecas universitárias, aderentes à iniciativa. O seu objectivo seria contribuir para um maior e mais fácil acesso aos catálogos das múltiplas bibliotecas, passando estes a serem percepcionados pelos utilizadores como um único e grande catálogo, onde poderiam proceder à pesquisa de obras, como se o fizessem apenas no catálogo da sua própria biblioteca. Após a pesquisa, poderiam proceder ao pedido de empréstimo de obras existentes em qualquer das bibliotecas, sendo esta facilidade implementada sobre as já existentes possibilidades de empréstimo entre bibliotecas.

Para além de fomentar um maior acesso à informação, este sistema teria também um efeito, não menosprezável, de contribuir para uma maior racionalização e optimização do investimento nos acervos das bibliotecas, podendo, por exemplo, evitar a compra por parte de múltiplas bibliotecas de obras menos requisitadas.

Com vista à materialização desta iniciativa, foi constituída uma comissão instaladora, com sede nos Serviços de Documentação da Universidade de Aveiro, da qual faziam parte alguns elementos do actual grupo de investigação do Laboratório de Sistemas de Informação e Telemática do IEETA. Esta iniciativa acabaria por ser descontinuada em finais de 1999, devido a constrangimentos de ordem logística, administrativa e financeira (RUBI, 1999).

Apesar da descontinuidade da iniciativa, o grupo de investigação referido antes considerou a ideia original da iniciativa – a implementação de um catálogo bibliográfico distribuído – meritória de todo o crédito. Por isso, e no âmbito do projecto "Memória de

África", este grupo decidiu colocar em prática a concepção e desenvolvimento dessa ideia, que consistia na implementação de um agregador de registos bibliográficos distribuídos, para ajuda à manutenção da biblioteca virtual nesse projecto. Como resultado, foi desenvolvido um protótipo, que permitiu avaliar da exequibilidade tecnológica de um tal sistema (Zagalo et al., 2000; Zagalo et al., 2001) e começou por ser o primeiro trabalho de investigação e desenvolvimento do presente projecto de doutoramento.

Depois desta primeira abordagem aos catálogos distribuídos, os interesses de investigação evoluíram no sentido das bibliotecas digitais, com o intuito de não só oferecer acesso a referências bibliográficas, mas oferecer também acesso às próprias obras. Dentro desta perspectiva, iniciou-se então a pesquisa e documentação do estado da arte, sobre a teoria da modelação e organização da informação nas bibliotecas digitais, assim como os sistemas e tecnologias que poderiam servir de base a tais sistemas. Foi concebido um modelo e uma arquitectura para uma biblioteca digital distribuída, na qual foi incluído de novo o acesso a catálogos bibliográficos distribuídos, perfazendo assim um catálogo bibliográfico virtual, no seio da biblioteca digital.

Para a concepção do modelo da biblioteca digital distribuída foram reutilizados alguns dos conceitos subjacentes à concepção do agregador de registos, nomeadamente, o aumento da interoperabilidade entre as diferentes fontes de informação (sistemas e dados); a integração da informação proveniente dessas fontes e proporcionar uma visão holística das diferentes fontes de informação.

## 1.4 A Relevância

A relevância deste trabalho subdivide-se em dois vértices, por um lado a relevância e impacto na sociedade que a solução proposta pode gerar e por outro a relevância ao nível tecnológico de uma solução de integração e resolução de problemas criados pelo elevado nível de diversidade.

Assim, no primeiro caso este projecto veio oferecer uma nova abordagem no acesso à informação por parte dos utilizadores, criando um modo de acesso único para todo um espectro de bibliotecas digitais. Ou seja, o utilizador passa a ter de perder menos tempo nas suas pesquisas porque já não necessita de visitar diferentes bases de informação, podendo fazê-lo a partir de um ponto integrado único. Esta pesquisa aumenta não apenas a velocidade como o rendimento da mesma, uma vez que passa a ser possível

comparar de imediato arquivos existentes em diferentes bibliotecas digitais e assim fazer a melhor opção em função das necessidades.

Na vertente tecnológica temos que a disponibilização de uma solução informática capaz de funcionar de modo independente com os diferentes protocolos entretanto criados para acesso e tratamento da informação vem libertar a escolha do protocolo, podendo assim cada arquivo digital optar pelo protocolo que mais lhe convém em função da informação de que dispõe. Para além disso e sendo uma solução de interoperabilidade que utiliza normas abertas, esta não afecta o curso natural das evoluções tecnológicas de cada um dos protocolos potenciando assim o desenvolvimento de cada um de modo independente. Nesta solução foram aplicados vários conceitos de base, como a distribuição e o paralelismo, não constituindo estes a novidade deste trabalho, mas antes a sua integração e aplicação.

Finalmente no campo da relevância e demonstrando alguma da eficácia da solução proposta temos a evidenciar que partes da proposta deste projecto de doutoramento foram, desde o seu início, aplicadas em diversos projectos de investigação nos quais participou o grupo de investigação do Laboratório de Sistemas de Informação e Telemática do IEETA - Instituto de Engenharia Electrónica e Telemática de Aveiro e ao qual o autor deste projecto de doutoramento pertence. Alguns desses projectos, são:

- o projecto “Memória de África” (MemAfrica, 2009) – projecto que visou inicialmente a criação de um repositório de referências bibliográficas, também chamado biblioteca virtual, sobre as inúmeras obras e documentos com a temática do desenvolvimento e cooperação com os PALOP - Países Africanos de Língua Oficial Portuguesa e da Lusofonia em geral. Mais tarde foram feitos desenvolvimentos para a criação de uma autêntica biblioteca digital, que passou também a disponibilizar um conjunto de obras digitalizadas, com a mesma temática. Este projecto iniciou-se em 1997, tendo estado em continuo desenvolvimento até à actualidade e viu inclusivamente o seu nome mudado para “Memória de África e do Oriente”, após os seus interesses terem sido alargados, em finais de 2008, a outros destinos fora de África;
- o projecto SinBAD - Sistema Integrado de Biblioteca e Arquivo Digital (SinBAD, 2007; Almeida, 2006) – projecto desenvolvido no seio da Universidade de Aveiro e financiado pelo programa Aveiro digital 2003 - 2006 (AvDigital, 2008), que teve por objectivo construir um sistema de informação capaz de armazenar e oferecer acesso aos diferentes tipos de documentos que são propriedade ou são produzidos na Universidade de Aveiro, como por exemplo livros, teses, dissertações,

fotografias, vídeos, músicas, etc. Um dos desafios deste projecto foi precisamente o de oferecer um acesso integrado, tanto aos conteúdos como às suas descrições, por forma a minorar tanto quanto possível o efeito de estes residirem em sistemas e formatos diversos;

- e os projectos "Debates Parlamentares" (DebParlamentares, 2009) e "Arquivo Histórico Parlamentar" (ArParlamentar, 2008) – projectos desenvolvidos, entre 2002 e 2004, para a Assembleia da República Portuguesa, que visaram proporcionar o acesso a informação parlamentar, histórica e actual, sob a forma escrita e audiovisual, em suporte digital e via Internet (Pinto et al., 2005).

## 1.5 A Estrutura

A dissertação aqui apresentada é fruto de um projecto de investigação aplicada que como tal reflecte sobre a sua própria estrutura essa natureza subdividindo-se em duas partes claras. A primeira parte, dedicada ao enquadramento e contextualização da investigação que abarca os capítulos 1, 2 e 3. A segunda parte dedicada à apresentação da concepção, desenvolvimento e teste, que se estende do capítulo 4 ao capítulo 7. No final, a dissertação termina com as conclusões e perspectivas futuras, no capítulo 8.

Desta forma, o primeiro capítulo serve a introdução ao trabalho, explicitando a motivação e relevância do problema tratado. O segundo capítulo enquadra o projecto no domínio das bibliotecas digitais, descrevendo algum estado da arte, de carácter geral e teórico. O terceiro capítulo passa em revisão o estado da arte de carácter mais específico e mais técnico.

Na segunda parte, o quarto capítulo faz a apresentação da plataforma de *middleware* proposta por este projecto. No quinto capítulo é descrito o Agregador de Registos Bibliográficos, concebido, desenvolvido e apresentado no seio deste projecto como um caso de estudo e demonstrador para a plataforma de *middleware* proposta. O sexto capítulo reflecte a concepção e desenvolvimento de *middleware* para o acesso distribuído de repositórios digitais de informação, conforme ao preconizado pela plataforma proposta. Finalmente, no sétimo capítulo, são apresentados os resultados de alguns testes efectuados, em cenários diferentes, sobre diversos elementos da plataforma.

No oitavo capítulo, conclui-se a dissertação, apresentando alguns dos aspectos que podem ser tidos como originais e evidenciando caminhos para o futuro da investigação neste domínio das bibliotecas digitais.

# Capítulo 2

# Bibliotecas Digitais

## 2.1 Introdução

Nos anos 30, o escritor de ficção científica, Herbert Wells, promovia o conceito de um "cérebro mundial", baseado numa enciclopédia mundial (Wells, 1937; Wells, 1938). Este serviria como suporte de memória à actividade mental de qualquer pessoa e teria a capacidade de crescer e modificar-se numa revisão contínua.

Uma década depois, Vannever Bush, um dos mais conceituados conselheiros do esforço de guerra dos EUA, inspirava ao desenvolvimento e utilização de uma máquina, para uso pessoal, à qual deu o nome de Memex e que seria uma espécie de ficheiro ou biblioteca privada, contendo todos os livros, registos e comunicações pessoais e cujo funcionamento completamente automatizado permitiria a consulta de qualquer informação com elevada velocidade e flexibilidade (Bush, 1945; Bush et al., 1991).

Duas décadas mais tarde, Licklider, chefe da secretaria de Técnicas de Processamento de Informação do Departamento de Defesa dos EUA, escreve um livro, intitulado "Bibliotecas do Futuro", onde discute os meios pelos quais a informação poderia ser armazenada e recolhida electronicamente (Licklider, 1965). Isto num momento em que ainda se encontravam em investigação possibilidades como a partilha e a operação directa do tempo de um processador, por parte dos utilizadores.

Estas personalidades, tidas como visionárias, despontaram o véu do futuro dos sistemas electrónicos de informação e foram uma fonte de inspiração para muitos investigadores que se dedicaram posteriormente à descoberta, concepção e desenvolvimento de tecnologias que possibilitaram a concretização de sistemas complexos de gestão de informação, como os existentes actualmente, e que são a base tecnológica das bibliotecas digitais.

As bibliotecas digitais surgem assim num contexto que visa sobretudo oferecer ao homem melhores condições para o acesso e salvaguarda da informação com o fim último de fomentar uma verdadeira sociedade do conhecimento.

## 2.2 Definições

Uma das mais compreensíveis definições de biblioteca digital que viu luz até ao momento foi forjada no decurso do *workshop* IEEE CAIA' 94 (Gladney et al., 1994):

> "Uma biblioteca digital é um agrupamento de meios informáticos, de armazenamento e de comunicações, conjuntamente com o conteúdo e software necessários para reproduzir, emular e alargar os serviços fornecidos pelas bibliotecas convencionais baseadas em papel e em outros meios de colecção, catalogação, busca e disseminação de informação. Uma biblioteca digital completa deverá fornecer todos os serviços essenciais das bibliotecas tradicionais e explorar também as bem conhecidas vantagens do armazenamento, pesquisa e comunicação digitais."

Esta definição apresenta a biblioteca digital como uma extensão da biblioteca tradicional à custa da utilização das novas tecnologias da informação e da comunicação. Esta perspectiva é partilhada por vários outros autores, como é o exemplo de Borbinha (Borbinha, 2000).

Contudo e volvida mais de uma década sobre o aparecimento das grandes iniciativas para a investigação e implementação de bibliotecas digitais, pode afirmar-se que não existe ainda uma definição única na qual todos os personagens, que fazem parte da grande comunidade que trabalha nesta área, se revejam.

Devido precisamente a este "separatismo" no modo de ver as bibliotecas digitais, Borgman recolheu um conjunto de definições oriundas dos mais diversos autores e conduziu um estudo de análise e apreciação para tentar compreender as razões das suas diferenças. A relevância deste estudo encontra-se na simples constatação de que, em geral e nos mais diversos domínios de investigação, o insucesso em definir

apropriadamente um termo, resulta invariavelmente no atraso do desenvolvimento da sua teoria, investigação e prática (Borgman, 1999).

Algumas conclusões do estudo:

- as diferenças nas definições espelham sobretudo diferenças de perspectiva;
- existem duas perspectivas predominantes: a dos investigadores, oriundos das ciências da computação e da informação, e a dos bibliotecários;
- os investigadores estão habituados a utilizar a palavra biblioteca em contextos muito diferenciados, como, por exemplo, um conjunto de funções para utilização em programação pode ser apelidada de uma biblioteca de funções;
- os bibliotecários, pelo seu lado, mantêm o significado da palavra biblioteca extremamente ligado ao sentido tradicional – uma sala ou um edifício onde, para além da existência das obras, são disponibilizados serviços aos utilizadores, como a ajuda da pesquisa de uma obra, por exemplo;
- as definições oriundas destas duas perspectivas, espelham mais as preocupações que cada uma das comunidades detém sobre a área, e acabam mais por servir para chamar a atenção, até de outras pessoas, para um conjunto específico de problemas e desta forma ser um catalisador para a sua resolução;
- apesar das tensões entre estas perspectivas, as respectivas comunidades não entraram em discussões directas sobre o assunto, preferindo simplesmente ignorarem-se;
- actualmente, mostram maior capacidade de cooperação, notando-se a sua participação conjunta em conferências da especialidade, sem que contudo o termo "biblioteca digital" tenha perdido a sua dualidade de significados.

Definição de biblioteca digital, segundo a comunidade de investigação:

> "Uma biblioteca digital é: (1) um serviço, (2) uma arquitectura, (3) um conjunto de recursos de informação, como bases de dados de texto, números, gráficos, áudio, vídeo, etc. e (4) um conjunto de ferramentas e capacidades para localizar, recolher e utilizar os recursos informativos existentes."

Esta definição foi uma das primeiras a surgir e foi utilizada nos *workshops* de preparação para a DLI 1 - *Digital Library Initiative Phase I*, tendo sido proposta pela própria Borgman, quando o termo utilizado ainda era "biblioteca electrónica" (Fox, 1993).

Definição de biblioteca digital, segundo a comunidade bibliotecária:

> "Bibliotecas digitais são organizações que providenciam os recursos, incluindo o pessoal especializado, para: seleccionar, estruturar, interpretar, distribuir, preservar a integridade e garantir a persistência das colecções de trabalhos digitais, de modo que estas estejam sempre disponíveis, de forma pronta e económica, para a utilização por uma comunidade definida ou um conjunto de comunidades."

Esta foi a definição adoptada pela DLF - *Digital Library Federation*, com o intuito de alcançar um entendimento comum do termo para todos os parceiros envolvidos na federação das suas bibliotecas digitais (Waters, 1998).

Com o objectivo de criar uma definição que fosse ao encontro dos ensejos das duas comunidades, a NSF - *National Science Foundation*, em 1996 e no decurso de um *workshop*, propôs uma definição mais abrangente, estruturada em dois pontos complementares (Borgman et al., 1996). Segue-se a sua reprodução:

> 1. "As bibliotecas digitais são um conjunto de recursos electrónicos e capacidades técnicas associadas para a criação, procura e utilização de informação. Neste sentido, estas são uma extensão e um aumento dos sistemas de recolha e armazenamento de informação que manipulam dados digitais em qualquer formato (texto, imagens, sons; imagens estáticas ou dinâmicas) e distribuídos na rede. O conteúdo das bibliotecas digitais inclui: dados, metadados, que descrevem vários aspectos dos dados (por exemplo: representação, autor, dono, direitos de reprodução) e metadados que consistem em ligações ou relações para outros dados ou metadados, sejam eles internos ou externos às bibliotecas digitais."
>
> 2. "As bibliotecas digitais são construídas, coligidas e organizadas por e para uma comunidade de utilizadores e as suas capacidades funcionais suportam a necessidade e utilização de informação dessa comunidade. Estas são um componente das comunidades nas quais indivíduos e grupos interagem uns com os outros, usando dados, recursos de informação e conhecimento e sistemas. Neste sentido, estas são uma extensão, aumento e integração de uma variedade de instituições de informação, tal como locais físicos onde recursos são seleccionados, coligidos, organizados, preservados e acedidos como suporte a uma comunidade de utilizadores. Estas instituições de informação incluem, entre outras: bibliotecas, museus, arquivos e escolas; mas as bibliotecas digitais também estendem e

> servem outro tipo de comunidades, como salas de aula, escritórios, laboratórios, casas e espaços públicos."

Como conclui Borgman, esta última definição estende o domínio das bibliotecas digitais pelas várias dimensões tecnológica, social e institucional (Borgman, 1999).

Deve ainda notar-se que nas chamadas para propostas no âmbito das iniciativas para as bibliotecas digitais nos EUA, como por exemplo na DLI 1 e na DLI 2, nunca foi utilizada uma definição explícita de biblioteca digital. Contudo, nas últimas chamadas a noção de biblioteca digital tinha evoluído e apresentava maiores preocupações com o papel social e institucional.

Para além do termo "biblioteca digital", outros termos como "biblioteca electrónica" e "biblioteca virtual" têm sido usados de forma mais ou menos indiscriminada sem uma definição explícita associada, pretendendo referirem-se ao mesmo. Contudo, alguns autores são de opinião de que o termo "biblioteca virtual", por exemplo, possui um significado mais restrito. Para esses autores, o termo "biblioteca virtual" é mais consistente com o conceito de catálogo electrónico de referências bibliográficas, que serve como um sistema de "apontadores" para material digital ou não, e não possui directamente qualquer obra para consulta (Pinto et al., 2000).

No âmbito do trabalho desenvolvido neste projecto de doutoramento, orientado sobretudo para a problemática da concepção e implementação tecnológica das bibliotecas digitais, existe uma associação, pelo menos empática, com as definições que espelham esse tipo de preocupações. Existe, mesmo, uma definição emanada do grupo de investigação de Stanford que é especialmente cara a este projecto (Reich and Winograd, 1995):

> "Uma biblioteca digital consiste numa colecção de serviços coordenados, baseados em colecções de materiais, podendo alguns deles não se encontrar directamente sob o controlo da organização que oferece o serviço no qual possuem um papel".

Esta definição simples e concisa, contextualizada pelos actuais serviços tecnológicos de informação, oferece ao presente trabalho uma excelente rede de suporte.

Sobre as restantes definições, as que aqui foram mencionadas e as que o não foram, este trabalho apresenta uma simples e despretensiosa reflexão: as definições sobre bibliotecas digitais, oriundas dos mais diversos quadrantes, não devem ser colocadas em contraposição mas devem ser vistas como perspectivas complementares, e por vezes até ortogonais. As múltiplas definições são muitas vezes oriundas de níveis de abstracção

diferentes, no tocante à visão do objecto "biblioteca digital". Para isso, pode estabelecer-se uma comparação, utilizando um objecto de massas como a televisão, por exemplo. Pode definir-se o termo televisão, do ponto de vista meramente tecnológico e nesse caso são tidos em conta apenas os atributos físicos e tecnológicos que visam à concepção e implementação desse objecto. Ou, de um ponto de vista comunicacional, num nível de abstracção em que se ignora a tecnologia quase por completo, podem ter-se em conta aspectos como os conteúdos a utilizar, o impacto social e psicológico desses conteúdos associados à possibilidade da sua comunicação a massas, etc. Como na área da televisão, a área das bibliotecas digitais apenas tem a ganhar com a multiplicidade de perspectivas existentes, pois vão com certeza contribuir para um cada vez maior engrandecimento da própria área e da sua visibilidade.

## 2.3 Características

Historicamente, as bibliotecas têm sido descritas como os repositórios do conhecimento e a sua filosofia de funcionamento tem colocado a tónica sobre as colecções, os serviços e os utilizadores.

Vários investigadores têm discutido as características específicas das bibliotecas digitais, podendo-se resumir as essenciais à seguinte lista (Lynch and Garcia-Molina, 1995; Cleveland, 1998; Arms, 2000; Soergel, 2009):

- podem conter uma variedade de recursos de informação digital, como texto, imagem, áudio e vídeo;
- reduzem largamente o espaço físico necessário para a salvaguarda dos recursos;
- os utilizadores podem encontrar-se distribuídos por qualquer parte no mundo;
- os utilizadores podem criar as suas próprias colecções de recursos, através de facilidades concedidas pelas bibliotecas digitais;
- fornecem acesso a vários tipos de recursos de informação que podem residir em diferentes servidores, em qualquer parte do mundo;
- vários utilizadores podem usar o mesmo recurso de informação ao mesmo tempo;
- trouxeram um novo paradigma, relativamente aos detentores dos recursos. Uma biblioteca digital pode não ser detentora de um determinado recurso de informação, mas poderá fornecer acesso a este, mediante a política de acesso a que este esteja sujeito: acesso livre ou acesso pago;
- devem possuir capacidades multilingue;

- devem fornecer melhores serviços de pesquisa e recolha de informação, assim como melhores mecanismos para a filtragem dessa informação;
- devem ser dotadas de capacidades para a preservação a longo termo (Gladney, 2006; Reis, 2009).

Algumas destas características encontram-se já amplamente implementadas, contudo algumas outras são ainda tema de investigação, no sentido de procurar oferecer cada vez melhores serviços.

## 2.4 Impacto

A construção e gestão de bibliotecas digitais envolve um elevado número de recursos financeiros e intelectuais. Até ao momento, já foram gastos em todo o mundo, centenas de milhões de euros em projectos de investigação neste domínio. Será, porventura, legítimo perguntar: porque é que as bibliotecas digitais devem ser construídas e em que é que estas vão ajudar as pessoas no quotidiano das suas vidas.

Arms fornece uma resposta: “As bibliotecas digitais estão a ser construídas na crença de que irão fornecer um melhor serviço de distribuição da informação do que era possível no passado” (Arms, 2000).

Esta resposta poderá ser percebida como demasiado simplista, parecendo mais um postulado de fé do que uma afirmação objectiva, fundamentada em requisitos bem especificados. Contudo, Arms fornece uma lista de benefícios, “mais palpáveis”, que as bibliotecas digitais poderão oferecer:

- a biblioteca digital leva a biblioteca ao utilizador – com a ajuda de um computador pessoal ligado à rede, a biblioteca poderá estar sempre em cima da secretária ou onde quer que o utilizador se encontre;
- o poder do computador é usado para pesquisar e explorar – apesar de os métodos utilizados pelos meios computacionais para a procura de informação não serem ainda os ideais, estes são, sem sombra de dúvida, uma grande ajuda na pesquisa de grandes volumes de informação;
- a informação pode ser partilhada – muitas bibliotecas possuem por vezes exemplares únicos de uma determinada obra. Numa biblioteca digital, esta, para além de ser acessível a múltiplos utilizadores, é preservada do constante manuseio e desgaste;

- a actualização da informação é fácil – muita informação, não possui um carácter permanente, necessitando de ser actualizada periodicamente e este processo é muito mais facilitado numa biblioteca digital;
- a informação está sempre acessível – a biblioteca digital não tem horário de abertura e fecho. O que não quer dizer que os sistemas computacionais ou de rede sejam infalíveis;
- possibilidade de novas formas de informação – a impressão não é sempre a melhor forma de registar e disseminar a informação. Por exemplo, existe informação que faz mais sentido encontrar-se numa base de dados a qual pode ser pesquisada ou analisada por computador e criarem-se diferentes perspectivas da mesma. A palavra escrita também é diferente da palavra pronunciada: a informação que se pode retirar de uma ou de outra poderá ser mais completa ou até diferente;
- possibilidade de maior colaboração – alguma investigação na Universidade da Califórnia, em Berkeley, demonstrou que as bibliotecas digitais potenciam a colaboração entre utilizadores (Wilensky, 2000). Esta colaboração irá ter um profundo impacto no ciclo de vida da informação académica, nomeadamente no processo através do qual investigadores, professores e alunos criam, utilizam e distribuem informação.

Outros fundamentos, ou as mais diversas razões, poderão presidir à decisão da implementação de bibliotecas digitais, contudo a curta lista de benefícios apontada acima preconiza, a médio ou longo prazo, um forte impacto social e económico, ultrapassando o mero impacto tecnológico. O que poderá levar a que no futuro se venham também a verificar as chamadas leis económicas da disrupção tecnológica (Downes and Mui, 2000), para o domínio das bibliotecas digitais.

## 2.5 Bibliotecas Existentes

Seguidamente são apresentados alguns exemplos de bibliotecas digitais existentes, com um simples objectivo ilustrativo.

As bibliotecas seleccionadas para integrar este conjunto, são exemplos que podem ser considerados referências, cada um no seu domínio ou contexto. Desta forma, foram seleccionadas bibliotecas representativas: das primeiras bibliotecas implementadas; de índole governamental; de índole nacional; pertencentes a organizações; e originárias em projectos de investigação.

### 2.5.1 Projecto Gutenberg

O projecto Gutenberg foi criado em Julho de 1971, por Michael Hart, com o objectivo de disponibilizar uma versão electrónica livre de livros de literatura que se encontram no domínio público dos EUA (Gutenberg, 2009). Este projecto é considerado como tendo sido (Lebert, 2008):

- o primeiro serviço de informação disponibilizado na Internet, quando esta ainda se encontrava em estado embrionário;
- o criador do livro electrónico;
- e, por conseguinte, a mais antiga biblioteca digital.

Apesar de ser alvo de alguma desconfiança, quanto à sua escalabilidade, por parte dos seus críticos, este projecto conta actualmente, com mais de trinta mil livros disponíveis e com dezenas de milhar de transferências por dia.

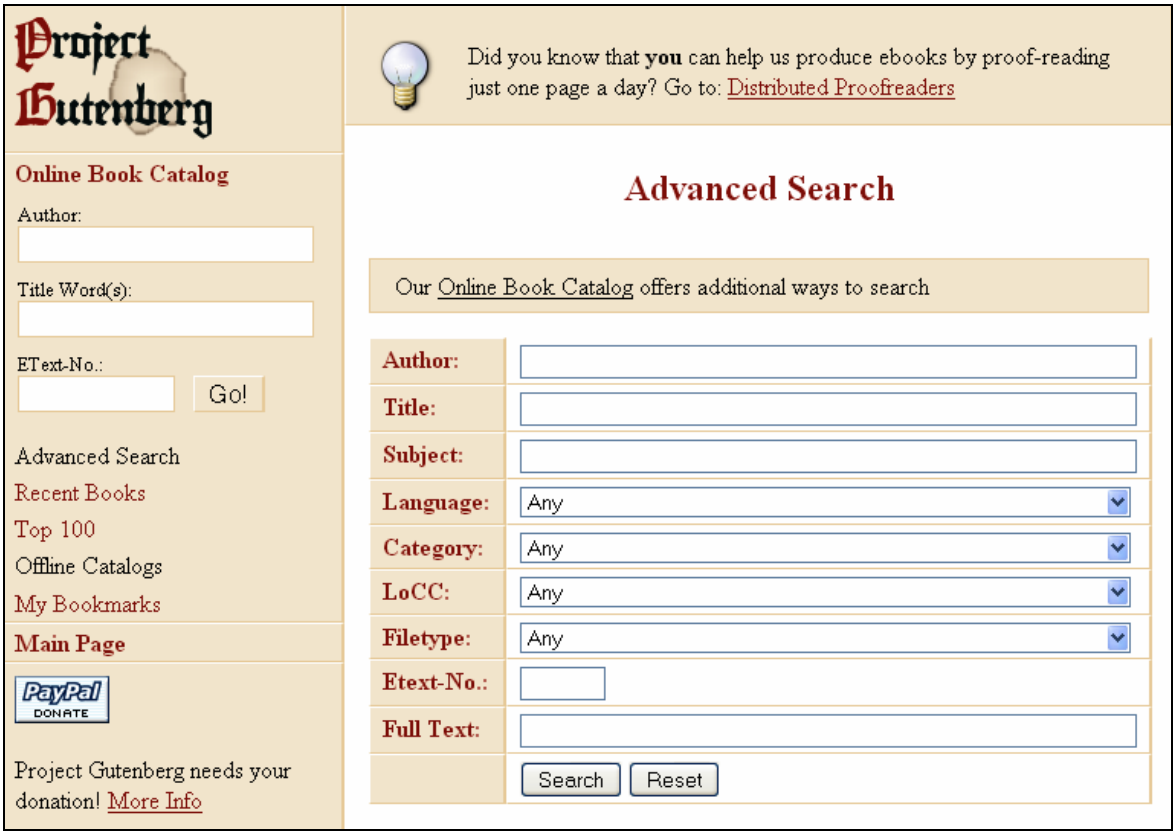



Figura 2.1 – Página web de entrada no Projecto Gutenberg (Gutenberg, 2009).

Na Figura 2.1 encontra-se uma imagem da página web que permite a pesquisa avançada do conjunto de livros que fazem parte do acervo do projecto Gutenberg.

O formato de armazenamento dos livros é o texto simples, utilizando o código ASCII. A adopção deste formato proporcionou, desde o início, um elevado nível de interoperabilidade ao nível dos dados, permitindo o acesso e utilização dos textos a partir de várias plataformas de hardware e software. A escalabilidade é conseguida através da distribuição e replicação dos livros por múltiplos pontos, em diferentes países, e milhares de voluntários que colaboram e contribuem na elaboração dos livros electrónicos.

Presentemente, com o intuito de oferecer uma melhor apresentação das obras, estas encontram-se também disponíveis no formato XHTML (W3C, 2002). Tanto um formato como outro podem ser acedidos na sua forma original ou na forma comprimida. Os protocolos disponíveis para transferência são: o HTTP (NWG, 1999); o FTP (NWG, 1985); e protocolos P2P (Schoder et al., 2005), cujas aplicações suportem ligações baseadas no *Magnet URI scheme* (Mohr, 2002).

Esta biblioteca digital assume uma enorme relevância, como projecto, devido à rara presença de duas características conjugadas:

- a primeira, é simplesmente o facto de ter sido a primeira biblioteca digital a tomar forma;
- a segunda, é o facto de ainda se manter actualmente em funcionamento e com tendência permanente para crescer, tanto em dimensão como em utilização.

De facto, é comum o aparecimento de projectos pioneiros e inovadores, que depois da fase de protótipo, não avançam para a fase de produção e, muitas vezes, quando o fazem não ganham a dimensão necessária à sua sustentação. Por estas razões, a biblioteca digital do Projecto Gutenberg merece um lugar de grande destaque entre as suas congéneres.

### 2.5.2 THOMAS

THOMAS é uma biblioteca digital de índole governamental, que faz parte da biblioteca do congresso dos EUA (LoC - *Library of Congress*) (LoC, 2009a).

Esta biblioteca digital viu o seu início de actividade em Janeiro de 1995, no princípio do centésimo quarto congresso, e desde esse momento tem vindo a armazenar e a permitir o acesso, ao público em geral, a um vasto conjunto de informação produzido pelo congresso dos EUA. Essa informação consiste, genericamente, em informação legislativa federal, como: propostas de lei, leis aprovadas, sessões do congresso, intervenções dos congressistas, documentos históricos, como os debates no congresso entre 1774 e 1873,

etc. De forma genérica pode afirmar-se que a biblioteca digital THOMAS permite a qualquer pessoa seguir em detalhe, dia a dia, a actividade do congresso dos EUA.

Actualmente, devido às recentes iniciativas promovidas no âmbito do *e-Government* (Palvia and Sharma, 2007), muitos países têm disponibilizado, aos seus cidadãos, o acesso a múltiplos serviços públicos por via electrónica. Um desses serviços é precisamente a possibilidade de consulta de informação referente aos trabalhos governativos e parlamentares do estado. Para além da mera disponibilidade de informação, este serviço pretende também aproximar mais o cidadão comum do processo governativo e legislativo que gere o seu país. No cumprimento deste objectivo existem já múltiplos sistemas de informação disponibilizados pelos estados, na forma de sites ou portais web e na forma de bibliotecas digitais.

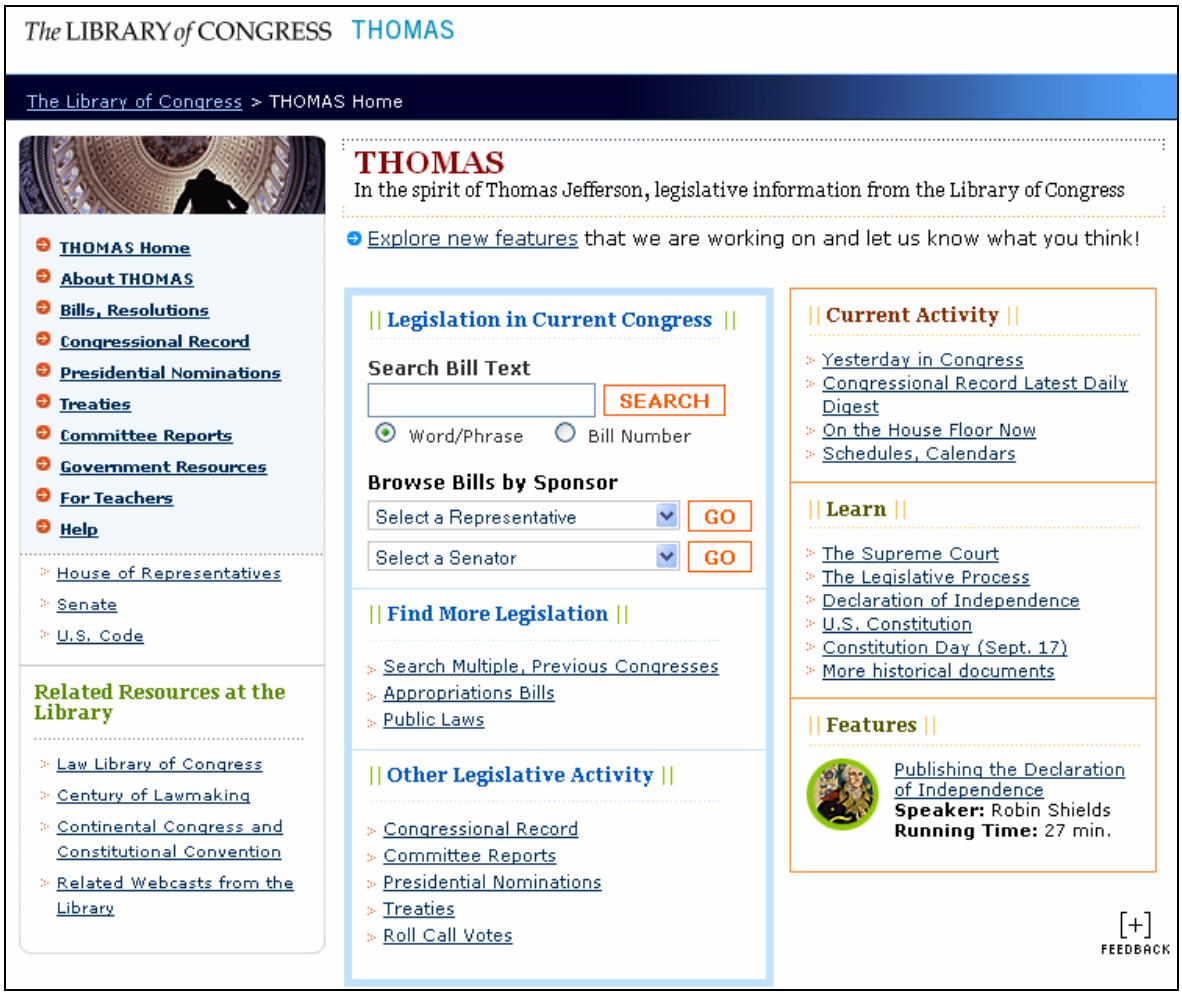



Figura 2.2 – Página web de entrada da biblioteca digital THOMAS (LoC, 2009a).

Na Figura 2.2 encontra-se uma imagem da página web de entrada na biblioteca digital THOMAS.

A presente dissertação refere especificamente a biblioteca digital THOMAS, porque esta consiste numa referência neste domínio ao precisamente servir de inspiração ao aparecimento de outras. De facto, foi uma das primeiras bibliotecas digitais com este propósito; permite o acesso a inúmera informação do congresso dos EUA, com actualização diária; e, mais uma vez, ao contrário de muitos outros projectos muito bem intencionados, não foi votado ao esquecimento, antes pelo contrário como se constata pela admirável periodicidade com que é actualizada.

### 2.5.3 ACM

A biblioteca digital da ACM - *Association of Computing Machinery* (ACM, 2009) é uma biblioteca de índole científica, mais especificamente no domínio das ciências e da engenharia da computação, e é pertença de uma organização: a ACM. Tem por isso um público-alvo muito mais restrito que as bibliotecas mencionadas antes.

Esta biblioteca confere acesso a:

- todos os artigos, incluindo o texto, dos vários jornais, revistas e actas de conferência da ACM, publicados nos últimos 50 anos;
- e índices de referências, incluídas nos artigos publicados pela ACM e outras editoras associadas, ascendendo neste momento a mais de 750 mil itens.

A pesquisa nesta biblioteca digital pode ser efectuada por qualquer utilizador. Os resultados obtidos podem incluir resumos e conjuntos de citações dos artigos encontrados. Contudo, o acesso ao conteúdo dos artigos encontra-se acessível apenas a subscritores.

Para além da funcionalidade de pesquisa, esta biblioteca oferece ao utilizador também a facilidade de exploração dos diversos textos disponíveis, por tipo de publicação: jornal, revista, *transactions*, boletins de grupos de interesse, actas de conferências e publicações de organizações afiliadas.

Na Figura 2.3 encontra-se uma imagem da página web de entrada na biblioteca digital da ACM.

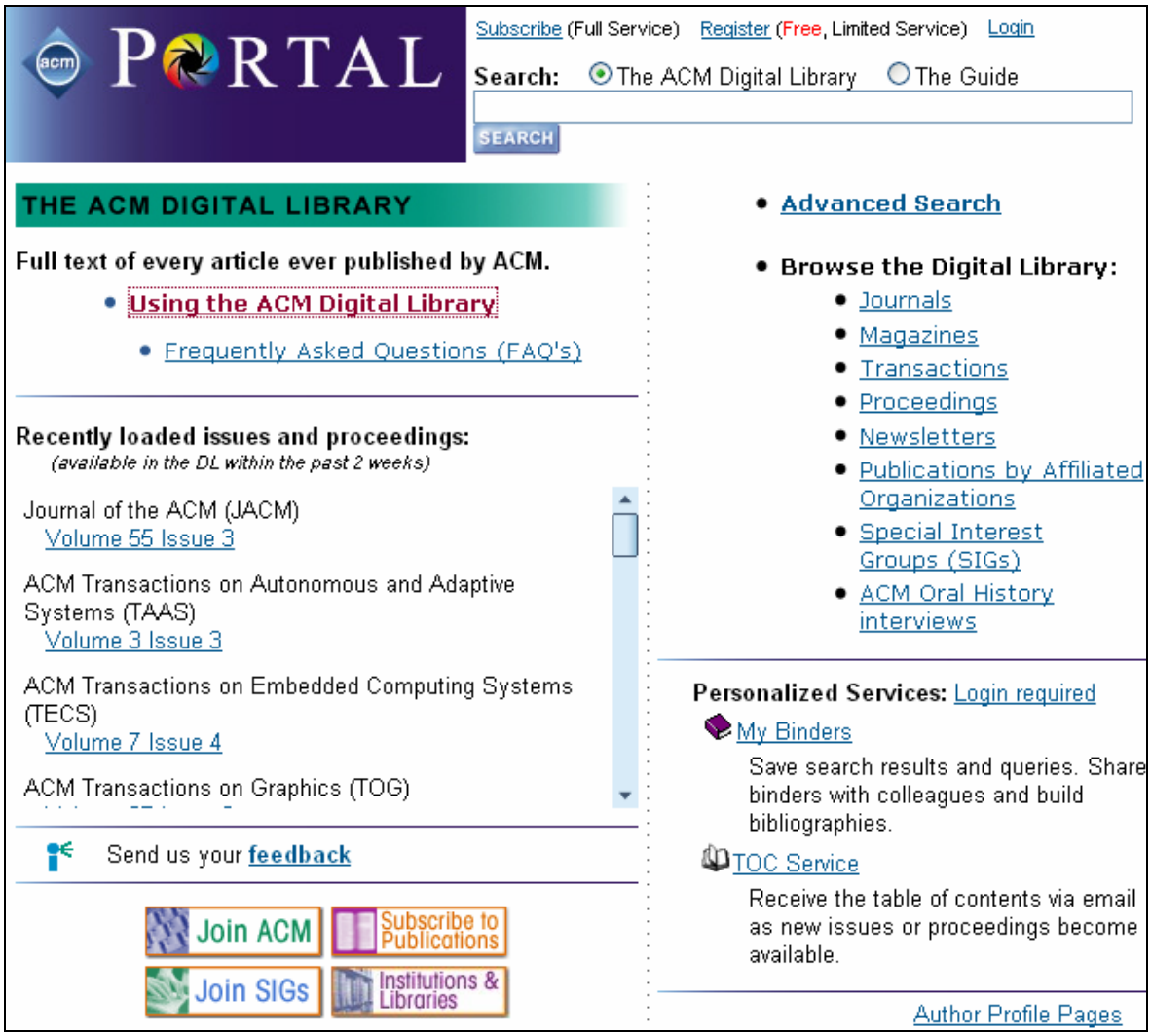



Figura 2.3 – Página web de entrada na biblioteca digital da ACM (ACM, 2009).

### 2.5.4 Biblioteca Digital da Nova Zelândia

> "O projecto da Biblioteca Digital da Nova Zelândia consiste num programa de investigação, com sede na Universidade de Waikato, que tem por objectivo o desenvolvimento de tecnologia de base para a implementação de bibliotecas digitais e disponibilizá-la publicamente de forma a que outros a possam utilizar para criar as suas próprias colecções" (NZDL, 2009).

Esta biblioteca digital (Figura 2.4) é originária de um projecto de investigação e oferece acesso a um vasto conjunto de colecções digitais com temáticas tão diversas como o registo de histórias de vida, contadas na primeira pessoa, de quem sofreu os efeitos de

desastres naturais até livros e documentos que pretendem contribuir para o desenvolvimento da humanidade, contendo informação prática sobre como reduzir a pobreza, aumentar o potencial humano e oferecer uma educação prática e útil para todos. Passando por temas como a literatura, as ciências computacionais, a ajuda médica e de socorro, etc.

Esta biblioteca digital conta com o apoio da UNESCO - *United Nations Educational, Scientific and Cultural Organization* (UNESCO, 2009) e da *Human Info NGO* (HINGO, 2009), duas organizações mundiais com preocupações principais no domínio da educação para o progresso da humanidade.

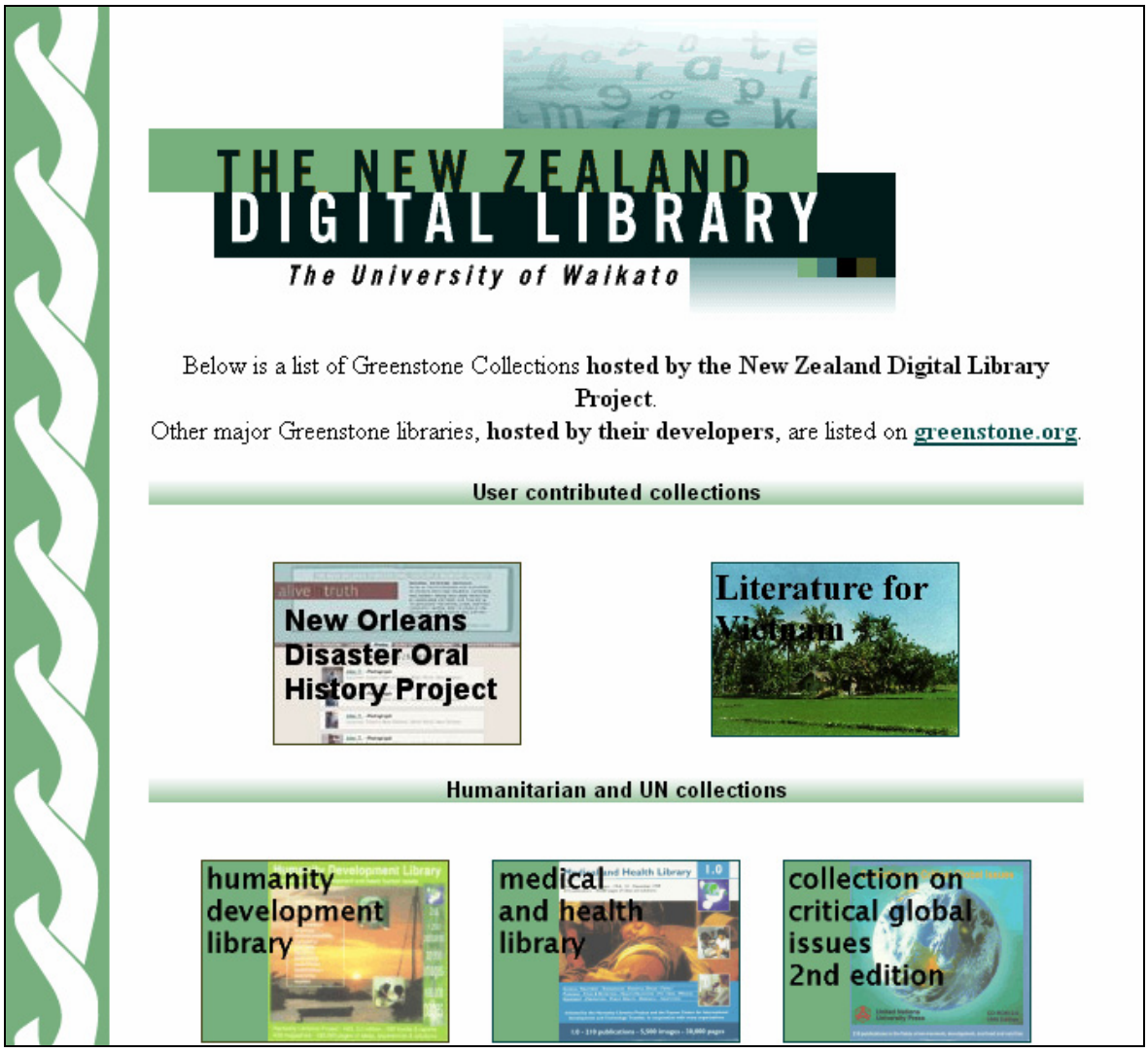



Figura 2.4 – Página web da Biblioteca Digital da Nova Zelândia (NZDL, 2009).

Apesar do seu conjunto de colecções muito diversificado, a maior relevância desta biblioteca digital assenta essencialmente no facto de ter como objectivo principal: a implementação e disponibilização livre de software que permite a outros a implementação das suas próprias bibliotecas digitais. O que faz com que a sua ambição vá para além da mera disponibilização de informação, pretendendo também ser um catalisador para que outros o façam.

### 2.5.5 Biblioteca Nacional Digital

A BND - Biblioteca Nacional Digital (Figura 2.5) consiste numa biblioteca digital, de índole nacional, criada e integrada na BNP - Biblioteca Nacional de Portugal (BND, 2009). Esta biblioteca digital tem por missão disponibilizar em linha o acesso à cópia digital de obras pertencentes às colecções da BNP.


Figura 2.5 – Página web de entrada na Biblioteca Nacional Digital (BND, 2009).

Nem todas as obras da BNP se encontram disponíveis para consulta na BND. Desde a criação desta biblioteca digital que a BNP definiu critérios para a selecção das obras a digitalizar e a disponibilizar em linha. No contexto da preservação de documentos de natureza frágil e de difícil manuseamento, têm sido seleccionadas, sobretudo, obras iconográficas e de material cartográfico. No contexto da valorização e divulgação do património documental nacional, têm sido seleccionados os documentos que apresentam elevado interesse histórico-cultural, tendo em conta a sua proveniência nacional, a respectiva data de publicação e a sua tipologia.

Na aplicação destes critérios, as obras digitalizadas são maioritariamente do tipo iconográfico e cartográfico, tendo sido digitalizados uma quantidade considerável de cartazes, estampas, desenhos, etc. A percentagem de obras digitalizadas do tipo textual é neste momento de cerca de 16% e é objectivo da BND aumentar o seu número. Até ao momento, foram digitalizados cerca de 400 títulos do livro antigo português (documentos impressos e publicados entre 1500 e 1800) e 1100 títulos dos séculos XVI e XVII, que vão gradualmente sendo colocados em linha.

A língua predominante das obras digitalizadas é o português, contudo podem-se encontrar também obras em francês, inglês, castelhano, italiano e alemão.

Os principais domínios representados são as artes, a história e a geografia. Embora em menor escala, encontram-se também representados os domínios das Ciências Sociais, Ciências Aplicadas, Literatura e Linguística.

O acesso às obras é feito através da navegação dos índices da BND ou através da pesquisa no catálogo bibliográfico.

### 2.5.6 Europeana

Em 28 de Abril de 2005, seis chefes de estado e de governo europeus enviaram um ofício à Presidência do Concelho e à Comissão Europeias, sugerindo a criação de uma biblioteca virtual europeia, para tornar os recursos culturais e científicos europeus acessíveis a todos.

Em resposta, a Comissão Europeia fez um comunicado, no dia 30 de Setembro do mesmo ano, intitulado “i2010: Bibliotecas Digitais” onde anunciou a sua estratégia para promover e suportar a criação de uma Biblioteca Digital Europeia. O objectivo da Comissão Europeia com esta biblioteca digital era tornar os recursos de informação europeus fáceis de utilizar num ambiente em linha. Isto, com base na herança cultural

rica da Europa, combinando ambientes multiculturais e multilingues com avanços tecnológicos e novos modelos de negócio.

A estratégia da Comissão Europeia tomou forma através do projecto Europeana (Europeana, 2010). O projecto Europeana desenvolveu, até ao momento, uma biblioteca virtual, em estado beta, que oferece o acesso, através de um ponto único de pesquisa, a obras residentes em múltiplas bibliotecas, arquivos e museus digitais europeus (Figura 2.6). O que pode ser perspectivado como uma Biblioteca Digital Europeia.

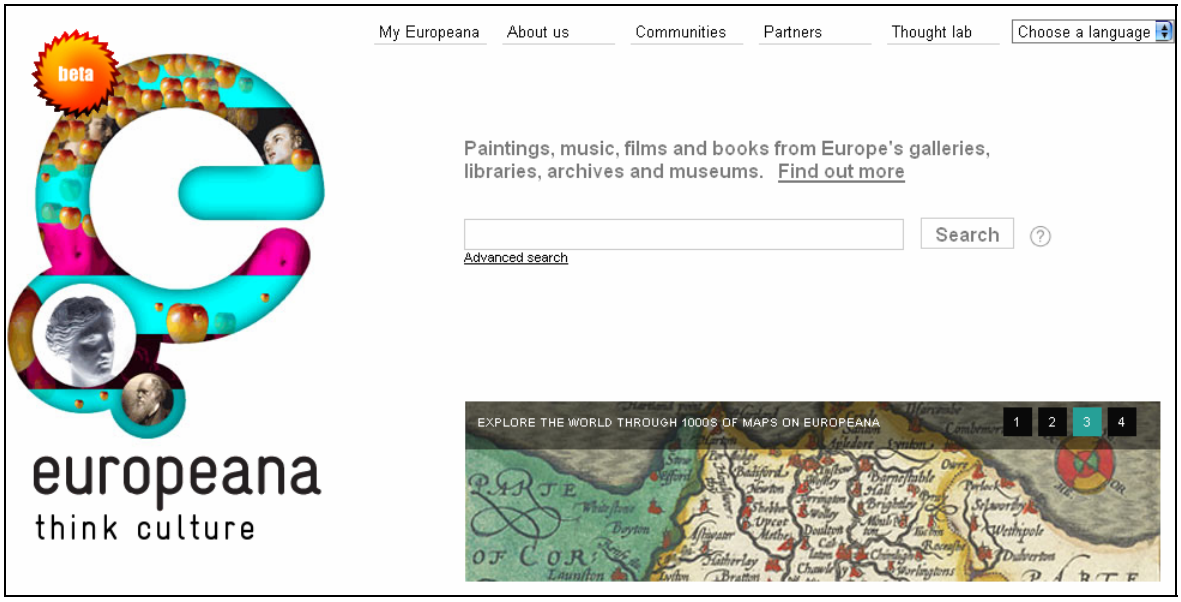



Figura 2.6 – Página web de entrada na Europeana (Europeana, 2009).

Na utilização da biblioteca virtual Europeana, as pesquisas são efectuadas no catálogo desta e posteriormente o acesso aos recursos digitais é efectuada acedendo directamente às bibliotecas ou arquivos digitais que os detêm. Por exemplo, após a pesquisa por uma obra, a Europeana disponibiliza informação descritiva acerca da obra e uma ligação a esta, que o utilizador deverá seguir caso a pretenda utilizar: ler um livro, ouvir uma música, visionar um filme, etc. Presentemente a Europeana oferece ligação a cerca de 6 milhões de objectos digitais. É desejo do projecto conseguir a ligação a 10 milhões de objectos no Verão de 2010 e o dobro desse número em 2012.

A sede da Europeana encontra-se na Biblioteca Nacional da Holanda e a sua supervisão está a cargo da EDL Foundation - *European Digital Library Foundation*, uma fundação criada com o objectivo de promover a colaboração entre museus, arquivos e

bibliotecas por forma a oferecer aos seus utilizadores um acesso integrado aos seus conteúdos.

## 2.6 Programas e Projectos

Nos finais da década de 1980, o Parlamento Europeu começou a dar atenção política à importância das bibliotecas na comunidade, reconhecendo-as como forças vitais no mercado da informação e como intermediárias para o conhecimento e a cultura.

No início da década de 1990 surgem então diversos projectos internacionais de investigação, no domínio das bibliotecas, que tinham por missão estudar e implementar novos recursos electrónicos, assim como fazer rentabilizar os já existentes, para oferecer serviços de mais-valia aos utentes. Um exemplo dos primeiros serviços a disponibilizar foi a consulta em linha dos catálogos bibliográficos das bibliotecas (OPACs - *Online Public Access Catalogs*).

Foi neste âmbito e no seguimento de projectos bem sucedidos, sobretudo relacionados com os registos bibliográficos, que emergiram projectos e iniciativas para a investigação das condições e tecnologias para a concepção e implementação de bibliotecas digitais.

Nos EUA, as bibliotecas digitais foram designadas, por esta altura, como uma “área de aplicação de desafio nacional” e como um “componente chave” para a infra-estrutura nacional de informação (Borgman, 2000). Foi, por isso, lançada uma iniciativa de investigação e desenvolvimento de grande envergadura, neste país, no domínio das bibliotecas digitais, a DLI - *Digital Library Initiative*, que decorreu em duas fases: a DLI 1 e a DLI 2.

Também na Europa foi aprovado um plano de acção para bibliotecas como uma das áreas do programa *Telematics*, no contexto dos terceiro e quarto programas quadro de financiamento para a investigação e desenvolvimento tecnológico, financiados pela União Europeia. Após este plano e no âmbito dos quinto, sexto e sétimo programas quadro de financiamento, a investigação sobre bibliotecas digitais continuou a desenrolar-se sob o domínio DigiCult - *Digital Heritage and Cultural Content*.

Apresenta-se a seguir um sumário de algumas das iniciativas e projectos internacionais de investigação no domínio das bibliotecas e bibliotecas digitais, que se considera serem das mais relevantes para a compreensão dos problemas e desafios colocados ao desenvolvimento de tais infra-estruturas informáticas.

### 2.6.1 DLI 1

A DLI 1, primeira fase da iniciativa DLI, decorreu entre 1994 e 1998 e foi patrocinada por três agências federais dos EUA: a NSF - *National Science Foundation*, a DARPA – *Defense Advanced Research Projects Agency* e a NASA - *National Aeronautics and Space Administration* (DLI1, 1998).

Nesta fase, a investigação foi principalmente orientada para a concepção e desenvolvimento de tecnologia, que permitisse a implementação das estruturas informáticas de sustento às bibliotecas digitais. Por isso, os investigadores envolvidos pertenciam principalmente à área das ciências e tecnologias da computação e da informação.

Os principais temas, sujeitos a investigação, foram:

- a captura de dados e metadados de todos os tipos (texto, imagens, som, voz, etc.) e a sua categorização e organização;
- a concepção e implementação de software e algoritmos para a pesquisa, a filtragem, a navegação, o resumo e a combinação de grandes volumes de dados;
- a utilização de bases de dados distribuídas;
- e a interoperabilidade entre serviços heterogéneos.

A título de exemplo, e porque versou sobre um tema que é caro ao trabalho apresentado nesta dissertação, refere-se de seguida, de forma sumária, o projecto *Interoperation Mechanisms among Heterogeneous Services*, sediado na Universidade de Stanford (Stanford, 2004).

Este projecto centrou a sua atenção na problemática da interoperabilidade entre colecções existentes e serviços relacionados com a publicação; e teve por objectivo conceber e implementar a infra-estrutura e os serviços necessários para a criação, disseminação, partilha e gestão da informação no contexto de uma biblioteca digital. Utilizando estas facilidades, os utilizadores poderiam ver as colecções e serviços electrónicos, oferecidos por terceiros, como recursos globais, que juntos compreenderiam uma biblioteca para uso na solução das suas tarefas (Paepcke, 1996).

O desafio na implementação de uma infra-estrutura que integrava repositórios e serviços já existentes era o facto de estes recursos já terem sido implementados de forma independente por diferentes fornecedores. Isto significava que estes recursos eram acedidos através de mecanismos muito diferentes, desde o nível dos protocolos de comunicação ao nível dos próprios modelos de acesso.

Ao nível da interface com o utilizador, deveria existir a maior transparência possível quanto aos problemas de interoperabilidade. Os utilizadores necessitavam de saber a origem da informação, contudo os detalhes de como a informação era recolhida de um repositório ou o modelo de interacção dos vários serviços particulares deveriam ser escondidos.

A arquitectura da infra-estrutura implementada neste projecto foi baseada na metáfora do bus de hardware, e por isso foi apelidada InfoBus (Paepcke et al., 2000), sugerindo o *plug-in* dos serviços, repositórios e clientes ao bus, para interagirem entre si utilizando a vantagem dos mecanismos de interoperabilidade aí existentes.

A implementação desta arquitectura utilizou a tecnologia CORBA - *Common Object Request Broker Architecture* (CORBA, 2008) e como implementação desta, a plataforma ILU - *Inter-Language Unification* (ILU, 1999), disponibilizada gratuitamente pela Xerox PARC.

Agregadas à infra-estrutura InfoBus, foi conseguida uma variedade de aplicações que ofereciam poderosas formas de pesquisa de informação, seja através de interfaces avançadas para a manipulação directa ou através da tecnologia de agentes autónomos. Este projecto conseguiu ainda um forte impacto no âmbito das questões legais e económicas em ambientes de rede.

### 2.6.2 DLI 2

A DLI 2, segunda fase da iniciativa, decorreu entre 1999 e 2004 e foi patrocinada por várias outras agências e instituições, para além das que estiveram presentes na DLI 1: NSF, DARPA, NASA, LoC - *Library of Congress*, NLM - *National Library of Medicine* e NEH - *National Endowment for the Humanities* (DLI2, 2004).

Nesta fase, para além dos aspectos tecnológicos, a investigação recaiu também sobre outros tipos de aspectos, como os sociais, os comportamentais e os económicos. Estes novos contornos da investigação aparecem no seguimento da definição de biblioteca digital, que ampliou o significado do termo, dada pela própria NSF e referida anteriormente na secção sobre definições.

Desta forma, a investigação, nesta fase, pode ser resumida e organizada em três áreas fundamentais, não sendo estas exaustivas:

- centrada no ser humano – tentar perceber o impacto e o potencial das bibliotecas digitais para melhorar as actividades humanas de criação, pesquisa e utilização da informação, assim como conceber soluções técnicas para o conseguir;

- centrada nas colecções e conteúdos – melhorar o entendimento do acesso avançado a novos conteúdos e colecções, como seja a eficiente captura, representação, preservação e arquivo de dados, a interoperabilidade de conteúdos e colecções, criar formas de lidar com aspectos sociais, económicos e legais associados à criação e utilização de colecções digitais, desenvolvimento e acesso de materiais educativos, etc.;
- centrada nos sistemas – orientada para os componentes tecnológicos e para a integração, por forma a conseguir a implementação de ambientes de informação dinâmicos, reactivos e capazes de adaptar corpos de dados amorfos, grandes e em expansão a estruturas e escalas definidas pelo utilizador.

O financiamento atribuído a esta fase foi de quase o dobro do da primeira e foi utilizado em mais de trinta projectos distintos, distribuídos por outras tantas universidades.

Esta fase da iniciativa incluiu fornecedores de conteúdos, entre os patrocinadores, para garantir uma base de trabalho sobre a qual os investigadores poderiam testar e validar novas tecnologias em ambientes centrados nas colecções e utilizadores. Com este objectivo a *Library of Congress* disponibilizou muitas das suas colecções da Memória Americana – um corpo substancial do seu conteúdo multimédia (LoC, 1998).

A LoC identificou ainda dez desafios que deveriam ter a atenção dos investigadores se estes realmente pretendiam criar, no século vinte e um, bibliotecas digitais de grandes dimensões e de grande eficácia. Esses desafios podem ser agrupados do seguinte modo:

*Construção dos Recursos*

- Desenvolvimento de tecnologia melhorada para a digitalização de materiais analógicos;
- Concepção de ferramentas para a pesquisa e recolha de informação que compensem a descrição e catalogação abreviadas ou incompletas;
- Concepção de ferramentas que facilitem o melhoramento da catalogação ou descrição, incorporando as contribuições dos utilizadores.

*Interoperabilidade*

- Estabelecimento de protocolos e normas para facilitar a criação de bibliotecas digitais distribuídas.

*Propriedade Intelectual*

- Ter em conta os cuidados legais associados ao acesso, cópia e disseminação de materiais digitais e físicos.

*Acesso Eficaz*

- Integração do acesso a ambos os materiais: digitais e físicos;
- Desenvolvimento de abordagens que possam apresentar recursos heterogéneos de forma coerente;
- Tornar útil a NDL - *National Digital Library* a diferentes comunidades de utilizadores e para diferentes propósitos;
- Fornecer ferramentas mais eficientes e flexíveis para a transformação de conteúdos digitais conforme as necessidades dos utilizadores.

*Manutenção dos Recursos*

- Desenvolvimento de modelos económicos para o suporte da NDL.

Estimulando o diálogo entre os implementadores, os fornecedores e os utilizadores finais de conteúdos digitais, a LoC esperava assim contribuir para estabelecer um círculo apertado de informação de retorno para partilhar as descobertas relativas à construção, manutenção, utilização e sustento das bibliotecas digitais.

### 2.6.3 Terceiro Programa Quadro

Sob o terceiro programa quadro, que decorreu entre 1990 e 1994, o primeiro programa de investigação para bibliotecas colocou a tónica no desenvolvimento de ferramentas e serviços inovadores, assim como nos recursos bibliográficos e infra-estruturas de rede para biblioteca, que são o suporte de tais serviços (FP3, 1994). O programa foi estruturado em torno de quatro linhas complementares de acção:

- bibliografias computorizadas – que pretendia criar, melhorar e harmonizar catálogos bibliográficos na Europa, compreensíveis pelo computador, e assim contribuir para a eficiência das bibliotecas e para a melhoria da partilha dos recursos entre elas;
- sistemas de rede e interligação de sistemas nas bibliotecas – que pretendia ajudar a construir serviços de rede entre bibliotecas, assegurando a investigação e a exploração de novas oportunidades concedidas pelos novos serviços de telecomunicações e pelo progresso dos sistemas OSI;

- serviços bibliotecários inovadores – que pretendia permitir às bibliotecas oferecer serviços bibliotecários, de custo mais efectivo, através do uso de tecnologias da informação e comunicação avançadas;
- produtos e ferramentas bibliotecárias baseadas na tecnologia – que pretendia fornecer um estímulo ao mercado europeu, encorajando o sector privado a trabalhar com bibliotecas para produzir produtos telemáticos comerciais e viáveis e serviços e ferramentas concebidas especificamente para bibliotecas.

Ao abrigo deste programa foram desenvolvidos vários projectos, entre os quais se mencionam os seguintes, a título de exemplo:

- MORE - *MARC Optical Recognition* – projecto com o objectivo de avaliar a possibilidade do reconhecimento óptico de caracteres (OCR) como abordagem para a conversão de catálogos bibliográficos, na forma impressa para a forma electrónica (MORE, 1994);
- ELISE - *Electronic library image service for Europe* – projecto que modelou um sistema para fornecer acesso a bancos de imagens a cores, como slides de objectos em museus, manuscritos ilustrados e material cartográfico, que se encontram em duas bibliotecas de dois países membros (ELISE, 1995);
- EUROPAGATE - *European SR-Z39.50 Gateway* – projecto com o objectivo de implementar um sistema *gateway*, baseado sobre o protocolo Z39.50, para permitir o acesso de utilizadores finais a catálogos bibliográficos, que utilizam diferentes normas de acesso (EUROPAGATE, 1996).

### 2.6.4 Quarto Programa Quadro

O segundo programa, implementado sob o quarto programa quadro, decorreu entre 1995 e 1998 e baseou-se nos resultados obtidos pelo programa anterior e actividades desse programa, ainda em curso (FP4, 1998).

No pressuposto de que, cada vez mais, a criação, distribuição, acesso e utilização da informação era feito de forma inteiramente electrónica, este novo programa elegeu as bibliotecas como participantes chave no movimento para uma infra-estrutura electrónica de informação.

Entendeu-se que a mais-valia das bibliotecas só poderia ser percebida através da criação de uma ampla infra-estrutura de bibliotecas a nível europeu e por isso, este programa, dirigiu a sua atenção para o suporte ao desenvolvimento de redes entre bibliotecas, fomentando a partilha e a optimização dos recursos e como meio de ligar

bibliotecas menos avançadas aos recursos e serviços de bibliotecas mais avançadas. Também foi incentivada uma maior aproximação orientada ao mercado, por parte das bibliotecas, assim como uma maior harmonização das práticas, predominantes nas bibliotecas do sector público, com as dos fornecedores de informação no sector privado.

Desta forma, o foco deste programa foi bastante mais ambicioso e mais integrador que o programa anterior e estruturou-se em torno de três linhas de acção:

- sistemas bibliotecários internos para utilização em rede – continuando o desenvolvimento de ferramentas que permitissem a adequação dos sistemas existentes ao ambiente de rede, o que pressupôs o desenvolvimento de sistemas para a aquisição, manipulação, conversão, disponibilização e gestão de uma variedade de formatos electrónicos, por forma que estes pudessem estar disponíveis através de sistemas telemáticos;
- sistemas telemáticos para cooperação entre bibliotecas e trabalho em rede – focando a atenção na passagem das bibliotecas do paradigma baseado em colecções para o paradigma orientado ao acesso, visto que a melhoria da cooperação entre as bibliotecas, fornecedores, editores e elas próprias, poderia aumentar significativamente o nível, a quantidade e a qualidade dos recursos e serviços a disponibilizar ao utilizador individual da biblioteca;
- serviços bibliotecários para o acesso a recursos de informação em rede – focou a mais valia e o papel de mediação das bibliotecas no mundo da informação em rede, visto considerar-se que as bibliotecas se encontravam bem posicionadas para desempenhar um papel importante na organização e distribuição de informação em rede e agirem como intermediárias entre o utilizador final e os recursos, desde que mobilizassem esforços para contribuir para os desenvolvimentos necessários.

Seguem-se dois exemplos de projectos desenvolvidos no quadro deste programa:

- BIBLINK - *Linking Publishers and National Bibliographic Services* – projecto que teve por objectivo desenvolver e melhorar os serviços nacionais de bibliografia, criando novas ligações entre os editores e as bibliotecas nacionais ou agências nacionais de bibliografia. Na primeira fase, foi investigado e desenvolvido um consenso sobre formatos e conteúdos de dados, assim como padrões de transmissão e identificação e autenticação de documentos. Na segunda fase, foi desenvolvida uma especificação técnica, completamente funcional, na forma de um demonstrador, que permitia a utilização dos formatos Dublin Core e SGML, como formatos de transferência com os editores, e produzir registos em formato

UNIMARC para conversão dos diferentes formatos bibliográficos nacionais (BIBLINK, 1999);

- CASA - *Cooperative Archive of Serials and Articles* – projecto com o objectivo de criar uma lista de periódicos baseada no sistema ISSN, que permitisse aos utilizadores a pesquisa e localização de periódicos nacionais em catálogos distribuídos, enquanto ofereceria a catálogos colectivos nacionais e agências ISSN a possibilidade de troca e revisão de informação bibliográfica dos periódicos, em cooperação com o centro internacional do ISSN (CASA, 1999).

### 2.6.5 Quinto Programa Quadro

No quinto programa quadro de financiamento para a investigação e desenvolvimento tecnológico, a União Europeia criou um novo domínio de investigação, que apelidou de DigiCult, integrado no programa IST - *Information Society Technologies*, e decorreu entre 1998 e 2002 (FP5, 2002).

A investigação no domínio DigiCult tinha por objectivo criar condições para a implementação de ferramentas e sistemas para a exploração de recursos pertencentes à herança cultural, que se encontrassem tanto na forma tradicional como digital. Ou seja, recursos criados como substitutos digitais dos objectos físicos originais ou recursos criados de origem na forma digital.

Baseando-se na premissa de que os recursos que compõem a herança cultural e científica são de fundamental valor para o presente e futuro da Europa, ambos como uma base única de conhecimento e como um enorme potencial de utilização comercial, o programa DigiCult orientou o trabalho de investigação para a necessidade de garantir que as instituições que detêm tais recursos possam explorar completamente as oportunidades criadas pelo advento das tecnologias digitais para fornecer um acesso de qualidade aos mesmos por parte dos cidadãos europeus, assim como preserva-los para futuro.

Os principais tópicos de investigação abordados por este programa foram:

- o suporte às bibliotecas digitais, distribuídas pela Europa, através da interligação e integração cultural e científica dos recursos digitais, para criar novos serviços e infra-estruturas; através da exploração e preservação física e digital dos artefactos, que vão desde manuscritos até filmes; e através do desenvolvimento de novos modelos de negócio com fim ao acesso e uso de recursos científicos e culturais;
- o aumento do acesso aos artefactos culturais e científicos que se encontram em museus, bibliotecas e arquivos, por parte do grande público, incluindo turistas e

crianças em idade escolar, através do uso de tecnologias inovativas, como dispositivos móveis, técnicas de digitalização e suporte à Internet;

- o desenvolvimento de novas formas de representar, vivenciar e preservar o passado, através da utilização de tecnologias de ponta, como a realidade virtual ou a visualização 3D, por exemplo; as suas aplicações seriam a reconstrução virtual, jogos educacionais interactivos em ambientes históricos reconstruídos, etc.;
- fornecer meios e competências a pequenos grupos de indivíduos, em comunidades locais, para a partilha e documentação de interesses, memórias e perspectivas comuns da sua herança local, construindo assim uma imagem "viva" da herança regional, através da Europa.

No âmbito do domínio DigiCult foi dado financiamento a mais de cem projectos de investigação, o que demonstra o elevado empenho da União Europeia na promoção da investigação e do desenvolvimento das tecnologias associadas às bibliotecas digitais. Neste quadro surgiu a iniciativa DELOS - *A Network of Excellence on Digital Libraries*, que veio oferecer um contexto para o desenvolvimento contínuo de uma agenda internacional de investigação e constituir um ponto de referência para projectos sobre bibliotecas digitais, financiados pelo programa IST, estimulando assim a troca de experiências nesta área multidisciplinar (DELOS, 2002). Esta iniciativa:

- criou ambientes de experimentação e facilitou a sua interoperabilidade;
- forneceu mecanismos para a avaliação de diferentes modelos, aproximações e técnicas;
- suportou a troca de componentes de software em código aberto (*open source*);
- contribuiu para a definição de padrões relevantes;
- desenvolveu modelos apropriados para iniciar a exploração das tecnologias de bibliotecas digitais pela indústria;
- criou uma rede de ligações entre a comunidade internacional de investigação;
- e organizou as actividades no âmbito de cinco fóruns: investigação, avaliação, padronização, disseminação e transferência de tecnologia e cooperação internacional.

### 2.6.6 Sexto Programa Quadro

O sexto programa quadro de financiamento para a investigação e desenvolvimento tecnológico deu continuidade ao domínio DigiCult, no qual se continuou a desenrolar a

investigação sobre bibliotecas digitais. Este programa decorreu entre 2002 e 2006 (FP6, 2006).

O programa anterior tinha colocado a tónica de investigação no contexto dos processos de aprendizagem e de acesso à herança cultural, optimizados através da tecnologia. No sexto programa, a tónica é colocada mais no contexto do acesso e preservação dos recursos culturais e científicos, com o objectivo de desenvolver sistemas e ferramentas para o suporte da acessibilidade e utilização desses recursos ao longo do tempo. Este trabalho foi conduzido em duas vertentes diferentes:

- a primeira, orientada para a complexidade emergente dos objectos e repositórios digitais de índole cultural e científica, através de um representação conceptual rica e de métodos avançados de acesso. O que implicava a criação de sinergias fortes com tecnologias de ponta para investigar e desenvolver aplicações direccionadas para comunidades específicas de utilizadores que utilizam, de forma pró-activa e criativa, conteúdos culturais em formas heterogéneas;
- a segunda, orientada para perceber como preservar a disponibilidade dos recursos digitais, ao longo do tempo, através de novos conceitos, técnicas e ferramentas. Esta vertente focou-se nas questões de investigação sobre como lidar com recursos que são complexos, dinâmicos, muito interactivos e de grande volume. Isto em trabalho experimental, a curto prazo, e em ambientes de produção e utilização, a longo prazo.

A iniciativa DELOS viu neste programa garantida a sua continuidade, tendo até este momento conseguido criar uma comunidade europeia muito activa na investigação sobre bibliotecas digitais (DELOS, 2007). Esta comunidade, através da publicação de mais de quinhentos artigos, forneceu significativos contributos para muitos componentes chave das bibliotecas digitais, tais como:

- arquitecturas avançadas e especializadas;
- captura automática de metadados e extracção a partir de colecções multimédia;
- mecanismos para a integração e automatização da inclusão e avaliação de material digital;
- ontologias para conceitos visuais e textuais;
- recolha, entrega e apresentação de informação de forma personalizada;
- interfaces amigáveis ao utilizador;
- serviços de anotação;

- e ambientes de experimentação para comparação e avaliação de sistemas.

Uma das actividades conjuntas da rede DELOS foi o desenvolvimento da geração seguinte de tecnologias para bibliotecas digitais, com o objectivo principal de criar serviços interoperáveis, que utilizam diferentes línguas e diferentes modos de funcionamento, e gestão integrada de conteúdos, ambos, para serem incorporados em sistemas industriais de gestão de bibliotecas digitais.

A par com esta iniciativa tiveram lugar múltiplos projectos de investigação dos quais se salienta o projecto BRICKS - *Building Resources for Integrated Cultural Knowledge Services*, pelo seu contributo para recriar o próprio conceito de biblioteca digital (BRICKS, 2007).

A ideia por detrás do projecto BRICKS foi a de desenvolver uma nova geração de bibliotecas digitais, cujo termo pudesse ser aplicado a museus digitais, arquivos digitais e outros tipos de sistemas de memória digital. No domínio tecnológico, foi baseado na plataforma para a Memória Digital Europeia; no domínio organizacional, foi suportado pela comunidade criada pelo projecto, que incluiu: fornecedores de conteúdos; profissionais, investigadores e estudantes do domínio das artes; e utilizadores interessados, em geral.

Este projecto foi estruturado em três áreas principais de trabalho:

- infra-estrutura – área orientada para o desenvolvimento tecnológico da plataforma para a Memória Digital Europeia, que consistiu num sistema de serviços em rede, baseado em normas abertas, capaz de integrar colecções heterogéneas de documentos digitais multimédia;
- aplicações – esta foi a área sobre a qual o projecto se centrou mais, com o objectivo de criar os serviços iniciais da Memória Digital Europeia. Estes serviços apresentam valor acrescido para as comunidades envolvidas e demonstram o potencial de mercado e fiabilidade da sua infra-estrutura. As aplicações oferecem um largo espectro de possibilidades para a exploração da herança cultural digital e podem ser usadas para o desenvolvimento de soluções integradas de comércio electrónico em organizações de herança cultural e para fornecedores e utilizadores de tecnologia;
- sustentabilidade – este foi outro importante objectivo do projecto, que consistiu no desenvolvimento de um espaço de criação, chamado *Factory*, que deveria ser auto-sustentável no futuro. Este espaço é orientado ao utilizador e ao serviço para suportar a partilha de conhecimento e recursos no domínio da herança cultural.

Com o objectivo de oferecer soluções inovadoras e personalizadas, foram consideradas as tecnologias mais avançadas como: a web semântica, os serviços web, os sistemas DRM - *Digital Rights Management*, as marcas de água, etc.

### 2.6.7 Sétimo Programa Quadro

No sétimo programa quadro de financiamento e no âmbito de um dos seus subprogramas – o programa ICT - *Information and Communication Technologies* – o domínio DigiCult continua a ser o contexto principal no qual se insere a investigação na área das bibliotecas digitais. Este programa de financiamento teve o seu início em 2007 e decorre actualmente até 2013 (FP7, 2007).

O primeiro programa de trabalho ICT definiu as prioridades de investigação para os anos 2007 e 2008, no qual a investigação sobre bibliotecas digitais e preservação digital fizeram parte do *Challenge 4 - 'Digital Libraries and Content'*. Essas prioridades foram: os objectos digitais, de índole cultural e/ou científico, em múltiplos formatos e provindos de múltiplas fontes, localizados em bibliotecas digitais distribuídas em larga escala através da Europa; e a assistência às comunidades na utilização criativa de conteúdos em contextos multidisciplinares e multilingues. O trabalho aqui desenvolvido teve por base: ambientes robustos e escaláveis; processos de digitalização, economicamente viáveis; facilidades de pesquisa semântica; e ferramentas para a preservação de conteúdos digitais.

O segundo programa de trabalho ICT definiu as prioridades de investigação para os anos 2009 e 2010, no mesmo contexto. Essas prioridades são:

- sistemas escaláveis e serviços para a preservação de diferentes tipos de recursos digitais, capazes de controlar todo o seu percurso, desde a sua criação ao seu arquivo;
- cenários avançados de preservação: métodos, modelos e ferramentas para a gestão de memória digital, focando-se em problemas que constituem desafios de preservação e que não são adequadamente tratados pelos actuais modelos;
- soluções inovadoras de montagem de bibliotecas digitais multimédia para a utilização cooperativa em contextos e comunidades específicas, melhorando a compreensão académica e as experiências digitais de herança cultural;
- experiências culturais adaptativas, explorando o potencial das tecnologias da informação e comunicação para a criação de perspectivas personalizadas de várias formas de expressão cultural;

- redes de investigação interdisciplinar, ligando domínios tecnológicos (como modelos computacionais, representação de conhecimento, visualização e gráficos), ciências da informação e da arquivística e ciências sociais e cognitivas;
- promoção da compreensão da investigação financiada pela Comunidade Europeia, possibilitando a abertura de novos serviços de índole cultural e de preservação da memória, baseados nas tecnologias da informação e comunicação, e potenciando o impacto das iniciativas nacionais associadas; identificação dos futuros grandes desafios; e estabelecimento de uma rede pan-europeia de centros de "memória viva".

No decorrer deste programa a iniciativa DELOS e o projecto BRICKS chegaram ao seu término. Os resultados alcançados pela iniciativa DELOS podem ser classificados em quatro categorias:

- transmissão de excelência e difusão de resultados da investigação às comunidades interessadas na sua aplicação, assim como preparação de novos investigadores em temas relacionados com as bibliotecas digitais;
- avanço do estado da arte num número de tecnologias cruciais para o desenvolvimento da geração seguinte das bibliotecas digitais;
- definição de um modelo de referência para as bibliotecas digitais;
- desenvolvimento do DelosDLMS, um protótipo para futuros sistemas de gestão de bibliotecas digitais.

A utilização da plataforma BRICKS, por seu lado, pode trazer vários benefícios para as instituições que a utilizam, como:

- a oportunidade de tomar parte em comunidades confiáveis e de valor acrescentado, nas quais a identidade, a responsabilidade e a qualidade dos actores envolvidos é certificada;
- a correcção e a legalidade dos produtos com conteúdos electrónicos é garantida;
- a validade dos procedimentos de licenciamento, das condições comerciais e das transacções é assegurada;
- a possibilidade de utilizar práticas e metodologias para medir o impacto dos investimentos em cultura digital e a atenção que os serviços e conteúdos recebem dos visitantes remotos; estes indicadores podem ajudar as instituições a definir estratégias de comércio electrónico.

Ainda em curso, encontra-se o projecto DL.org - *Coordination Action on Digital Library Interoperability, Best Practices, and Modelling Foundations* (DL.org, 2009). Este projecto tem como objectivo a criação de uma plataforma onde representantes chave das maiores iniciativas e projectos em curso, relacionados com as bibliotecas digitais, possam colaborar, discutir experiências, trocar competências, trabalhar na interoperabilidade das suas soluções, promover normas partilhadas e proporcionar à comunidade das bibliotecas digitais uma profunda compreensão das questões chave e novas direcções.

Este projecto utiliza o modelo de referência DELOS para bibliotecas digitais, como base conceptual e operacional, para de uma forma inovadora tentar atingir o objectivo proposto. O seu principal instrumento consiste em seis grupos de trabalho temáticos, compostos por parceiros do projecto e representantes de projectos e organizações importantes no domínio das bibliotecas digitais, de modo a atingir o máximo impacto na comunidade das bibliotecas digitais, assim como outras.

Como resultados, espera-se que este projecto consiga oferecer:

- um guia sobre tecnologias e metodologias em bibliotecas digitais, no qual figurem os actuais melhores exemplos e práticas para facilitar a interligação entre sistemas existentes e melhorar a mesma em questões críticas de interoperabilidade;
- uma versão consolidada e melhorada do modelo de referência DELOS;
- *workshops*, cursos de verão, cursos à distância e actividades de disseminação para comunicar o impacto dos resultados do projecto a comunidades relevantes.

### 2.6.8 i2010 DLI

A i2010 DLI - *Digital Library Initiative* (i2010DLI, 2009) é uma iniciativa da União Europeia no domínio das bibliotecas digitais, a par com a iniciativa congénere do sétimo programa quadro, que tem por objectivo colocar todos os recursos culturais e científicos europeus, como livros, filmes, mapas, fotografias, música, etc., acessíveis a todos e proceder à sua preservação para as gerações futuras.

Esta iniciativa centra-se em duas áreas:

- a herança cultural – criando versões electrónicas dos materiais existentes nas bibliotecas, arquivos e museus europeus; colocando-as acessíveis na rede para trabalho, estudo e lazer; e fazendo a sua preservação para a posteridade;

- e a informação científica – fazendo com que as descobertas científicas tenham uma maior acessibilidade, tanto na amplitude do seu conjunto como na amplitude do tempo de disponibilidade.

A chave para atingir estes objectivos encontra-se no desenvolvimento do projecto Europeana (Europeana, 2010). Como referido anteriormente, o projecto Europeana assume a forma de uma biblioteca virtual que permite a pesquisa e o acesso a recursos que se encontram distribuídos por bibliotecas, arquivos e museus digitais europeus, constituindo assim a Biblioteca Digital Europeia. Este projecto é suportado por uma rede temática, financiada pela Comissão Europeia no âmbito da i2010 DLI, e é composta por 100 representantes de organizações patrimoniais e científicas e especialistas em tecnologias da informação, de toda a Europa, que contribuem para o trabalho de concepção e desenvolvimento dos aspectos técnicos e de usabilidade.

A Europeana detém um catálogo centralizado de metadados acerca dos objectos digitais disponíveis para utilização. As pesquisas são efectuadas neste catálogo, contudo o acesso aos objectos digitais é efectuada nos sítios disponibilizados pelos seus proprietários. Este catálogo de metadados é também exposto aos motores de busca web, permitindo a pesquisa do que é comummente chamado de web profunda.

Com vista à obtenção da informação a incorporar no seu catálogo, a Europeana impõe alguns requisitos técnicos. Um deles é a normalização dos metadados segundo o esquema ESE - *Europeana Semantic Elements* (ESE, 2010a). Este esquema tem por base os quinze elementos do conjunto DC - Dublin Core (DCMI, 2008), aos quais adiciona ainda os seguintes: isShownBy, isShownAt, userTag, unstored, object, language, provider, type, uri, year, hasObject e country. Para este efeito, a Europeana disponibiliza um XML Schema do ESE (ESE, 2010b), que permite fazer a validação dos metadados recém-chegados.

Outro requisito é a necessidade da existência de um identificador persistente, como um URL, para cada um dos objectos digitais. O que permite a ligação entre os metadados, residentes na Europeana, aos objectos digitais residentes nos sites dos seus proprietários.

Entre os fornecedores de metadados à Europeana incluem-se organizações individuais, que detêm e fornecem directamente os metadados, e agregadores, que são organizações que recolhem informação de múltiplas organizações individuais, procedem à sua normalização e canalizam-na para a Europeana.

Um agregador actua como uma interface entre a Europeana e as organizações individuais que detêm a informação. A Europeana trabalha preferencialmente com agregadores, mais do que instituições individuais.

Os agregadores podem, ou não, estar acessíveis ao público. No caso em que não estão, são chamados de *dark aggregators*. Podem agir como simples intermediários, coleccionando apenas metadados e ligações para os objectos digitais, ou podem agir como repositórios, armazenando também os próprios objectos. Existem diferentes tipos de agregadores: nacionais, regionais, temáticos e de domínio (por exemplo bibliotecas, museus, arquivos, televisão, etc.). Em conjunto, estes agregadores levam uma larga massa crítica à Europeana, sobre os conteúdos provindos dos mais diversos fornecedores.

Algumas das funções de um agregador são:

- a recolha de informação sobre as organizações individuais e os seus sistemas de entrega;
- a recolha de informação sobre os recursos digitais (metadados) para servir de substituto (*surrogate*);
- a eliminação da duplicação e ambiguidade dos dados e enriquecimento dos mesmos com atributos significativos;
- a verificação da acessibilidade dos recursos digitais;
- a manutenção dos dados para que a Europeana os possa recolher.

Com vista à obtenção da interoperabilidade funcional entre a Europeana e as organizações de onde esta procede à recolha dos dados, é utilizado o protocolo OAI-PMH - *Open Archives Initiative Protocol for Metadata Harvesting* (OAI, 2008a), o qual permite a recolha de metadados das fontes distribuídas.

## 2.7 Revisão

Neste capítulo é feito o enquadramento teórico deste projecto de doutoramento, dando uma visão geral da área de investigação em que este se insere.

Começa-se por uma introdução aos conceitos fundamentais das bibliotecas digitais, como as suas várias definições e múltiplas características, mostrando o quanto este domínio é um domínio multidisciplinar.

De seguida, apresentam-se alguns exemplos de bibliotecas digitais, consideradas relevantes nos seus âmbitos particulares, com o objectivo principal de mostrar a evidência da sua exequibilidade.

Por fim, é apresentado um conjunto de iniciativas e projectos de investigação, que embora um subconjunto da totalidade, mostra bem o compromisso das instâncias governamentais em contribuir para o avanço do conhecimento sobre o desenvolvimento das bibliotecas digitais e para a sua implementação efectiva, por forma a oferecer ao maior número possível de pessoas a possibilidade de acesso a obras que em muitos casos só podem ser acedidas por um número bastante limitado.

# Capítulo 3

# Arquitecturas e Tecnologias

Este projecto de doutoramento tem como enfoque principal de investigação, a engenharia. Torna-se, por isso, clara a necessidade de analisar, assim como evidenciar e enfatizar as diferentes arquitecturas, tecnologias e abordagens por detrás do bom funcionamento das bibliotecas digitais.

Este capítulo apresenta e coloca à discussão alguns dos diferentes modelos, plataformas, protocolos de pesquisa e normas de metadados que são utilizados na implementação de bibliotecas digitais.

Como introdução a esta discussão é apresentado o conceito de catálogo colectivo, e mais especificamente o modelo de catálogo colectivo virtual, por se tratar de um conceito e de um modelo que precederam o conceito e modelos das bibliotecas digitais. A influência dos primeiros sobre estes últimos não pode ser menosprezada visto que ela sobressai principalmente aquando da tentativa de analisar e compreender o funcionamento das bibliotecas digitais distribuídas. Neste projecto, foi o estudo, a concepção e o desenvolvimento de um catálogo colectivo virtual que esteve na génese de todo o restante trabalho desenvolvido sobre as bibliotecas digitais.

## 3.1 Catálogos Colectivos Virtuais

Desde os anos setenta que muitas instituições, com largos repositórios distribuídos de informação bibliográfica, têm tido uma preocupação constante na unificação das suas fontes de recursos. A frustração sentida por muitos utilizadores nos seus esforços para descobrir fontes de recursos relevantes, aprendendo interfaces de utilizador específicas e usando linguagens de pesquisa e convenções de semântica inconsistentes, levou a que as instituições e comunidades bibliográficas se interessassem pelo desenvolvimento de sistemas que pudessem assim harmonizar as complexidades da paisagem informativa, caracterizada por inúmeros recursos diferentes (Payette and Rieger, 1997), numa paisagem mais clara e homogénea, ou seja, criando catálogos colectivos.

### 3.1.1 Catálogos Colectivos

Um catálogo colectivo de registos bibliográficos consiste num catálogo que representa a união de múltiplos e diferentes catálogos bibliográficos. O seu principal objectivo é o de oferecer ao utilizador uma perspectiva global sobre todos os registos bibliográficos pertencentes a um conjunto de bibliotecas, podendo este identificar uma obra e a sua localização através de uma única pesquisa, ao invés da necessidade de múltiplas pesquisas nos catálogos locais de cada uma das bibliotecas. As próprias bibliotecas locais beneficiam do catálogo colectivo através da possibilidade de partilha entre si da informação bibliográfica (Coyle, 2000).

A instituição pioneira por excelência neste campo foi o OCLC – *Online Computer Library Center*, fundado em 1967 com o nome *Ohio College Library Center* pelos presidentes das Universidades do estado do Ohio - EUA, com o objectivo de partilhar recursos e reduzir custos (OCLC, 2009a).

Outra instituição que demonstra, desde longa data, grande interesse nesta questão é a Universidade da Califórnia. Tratando-se de uma instituição que possui diversos campus distribuídos pelo estado, em 1977 esta Universidade adoptou a máxima: “Uma Universidade, Uma Biblioteca”. Inicialmente, através de algumas tentativas com um catálogo de livros e subsequente versão em microfichas, e depois em 1982 através da disponibilização on-line de um catálogo colectivo que permitia a consulta de registos bibliográficos oriundos de todos os campus da universidade, assim como de outras instituições que aderiram à iniciativa (Coyle, 2000).

Em ambas as instituições, foram implementados catálogos colectivos centralizados.

### 3.1.2 Catálogos Centralizados

Um catálogo colectivo centralizado é baseado numa base de dados centralizada que possui uma cópia de todos os registos bibliográficos constantes dos catálogos locais filiados. Este tipo de catálogo necessita, para além da sua criação, de uma tarefa periódica de actualização, o que implica a recolha, processamento e armazenamento dos registos, para que o seu conteúdo reflicta com fidelidade o conteúdo dos catálogos filiados (Wells et al., 1998).

Em virtude das exigências de manutenção colocadas pelos catálogos centralizados e do aparecimento de protocolos especialmente orientados para a pesquisa bibliográfica, como é exemplo o Z39.50 (ZIG, 2003), diversas instituições iniciaram estudos no sentido de avaliar a viabilidade de substituição dos seus catálogos centralizados por catálogos "virtuais". Exemplos de instituições pioneiras a enveredar esforços neste sentido foram a Biblioteca Digital da Califórnia (Coyle, 2000) e a Biblioteca Nacional do Canadá (Lunau, 1998a).

### 3.1.3 Catálogos Virtuais

Um catálogo virtual é criado em tempo real através da pesquisa simultânea a múltiplos catálogos distribuídos, criando assim uma imagem de catálogo único aos olhos do utilizador final. O catálogo colectivo virtual assume-se como uma alternativa ao catálogo colectivo centralizado, impondo menor esforço de manutenção ao catálogo colectivo, mas impondo maior esforço de processamento e conectividade deste com os catálogos distribuídos (Coyle, 2000).

Nos projectos CDL – *California Digital Library* e vCuc – *Virtual Canadian Union Catalogue* criados pelas instituições mencionadas acima foram identificados um largo número de aspectos relacionados com o uso do protocolo Z39.50 para a criação de catálogos virtuais, que demonstraram haver questões pertinentes para quais se deveria continuar a procurar soluções mais aceitáveis (Lunau, 1998a; Coyle, 2000). Em mais detalhe:

- A Concorrência das Sessões

  O protocolo Z39.50, apesar de orientado à sessão e permitir, na sua versão de 1995, a pesquisa concorrente, define apenas interacções entre duas máquinas (chamadas "Origem" e "Destino" na norma). O que levava à necessidade de desenvolver aplicações cliente que suportassem múltiplas e concorrentes sessões com diversos servidores;

- A Identificação e Remoção de Registos Duplicados

  Sendo perfeitamente natural que diferentes bibliotecas sejam detentoras de uma mesma obra, é evidente que a pesquisa a múltiplas bibliotecas tenha como resultado a recepção de múltiplas referências à mesma obra. Tal é gerador de um conjunto de registos bibliográficos duplicados, que necessita do processamento adequado para evitar a sobrecarga do utilizador com informação redundante;

- Consistência Semântica das Pesquisas

  Os servidores e bases de dados destino suportam uma grande variedade de atributos de pesquisa e combinações desses atributos, conformes ao protocolo Z39.50. Contudo é muito comum que diferentes implementadores, tanto de servidores Z39.50 como de bases de dados, atribuam diferente semântica aos mesmos atributos. Tal facto leva facilmente à inconsistência dos resultados de pesquisa. Uma medida preventiva para evitar esse resultado, é por exemplo utilizar apenas um subconjunto dos atributos de pesquisa que seja o denominador comum entre todos os sistemas a pesquisar;

- Ligações a Outras Aplicações

  Geralmente a pesquisa num catálogo colectivo distribuído é conduzida no contexto de outras operações: a referenciação, a catalogação, a pesquisa académica ou o empréstimo entre bibliotecas. Desta forma é necessário que o cliente Z39.50 seja capaz de transferir a informação recolhida de forma eficaz a outras aplicações.

Tendo por objectivo apresentar soluções para algumas destas questões e sobretudo aumentar o nível de interoperabilidade entre os sistemas Z39.50, foram desenvolvidos acordos, referidos como perfis (*profiles*), nos quais se definem conjuntos comuns de características que devem ser respeitados aquando da implementação e utilização dos sistemas (Lunau, 1998b). Como exemplo de alguns desses perfis, referem-se o ATS-1 (ZIG, 1997a), o ZDSR (ZIG, 1997b) e o Bath Profile (BathGroup, 2003), tendo este último aproveitado os esforços desenvolvidos por vários outros perfis precedentes. Para além destes perfis, outros foram desenvolvidos com diferentes aplicabilidades, contudo estes assumiram particular importância no desenvolvimento de alguns catálogos colectivos virtuais.

A título exemplificativo, referem-se de seguida dois catálogos virtuais: o catálogo ZZZ da PORBASE (Figura 3.1), sediado em território nacional, e o catálogo KVK (Figura 3.2), sediado na Alemanha.


Figura 3.1 – Página web de entrada no catálogo virtual ZZZ (PORBASE, 2006).

O catálogo ZZZ consiste num serviço disponibilizado pela PORBASE (Base Nacional de Dados Bibliográficos) que permite pesquisar simultaneamente em diversos catálogos bibliográficos de algumas bibliotecas portuguesas (PORBASE, 2006). Este serviço oferece uma interface ao utilizador baseada na web e efectua o acesso às bibliotecas

através do protocolo Z39.50. Os resultados são visualizados pelo utilizador nos formatos Dublin Core (DCMI, 2008) e UNIMARC (IFLA, 1999) e cada registo pode ser descarregado, individualmente, no formato UNIMARC / ISO 2709 (ISO, 2008).

Por sua vez, o catálogo KVK – *Karlsruhe Virtueller Katalog* é disponibilizado pela Universidade de Karlsruhe – Alemanha, e permite a pesquisa simultânea de catálogos bibliográficos pertencentes diversas bibliotecas distribuídas pelo mundo, desde países em toda a Europa até aos Estados Unidos e Canadá (ULK, 2006). À imagem do catálogo ZZZ, também este catálogo disponibiliza uma interface ao utilizador baseada na web e o acesso às diferentes bibliotecas, efectuado através do protocolo Z39.50 (Monnich, 2001).


Figura 3.2 – Página web de entrada no catálogo virtual KVK (ULK, 2006).

A visualização dos resultados pelo utilizador, contudo, é feita de modo diverso. Neste caso, os registos são mostrados na forma de título abreviado, podendo o utilizador aceder a maior detalhe ao clicar sobre estes. Ao executar essa operação, o utilizador é redireccionado para a página oficial da biblioteca que detém o registo, acedendo assim à informação que essa biblioteca particular disponibiliza.

Para além da consulta a catálogos de bibliotecas, o KVK permite também a consulta de catálogos electrónicos de livrarias. A forma de visualização é idêntica, contudo não se encontra documentada qualquer informação acerca do modo de acesso a estes catálogos.

## 3.2 Modelos

As bibliotecas digitais referem-se por norma a sistemas de informação de âmbito heterogéneo e com funcionamento diferenciado. Sistemas que vão desde os objectos digitais, sistemas de referências, conteúdos administrativos, conteúdos complexos de investigação e repositórios de metadados. Toda esta heterogeneidade de informação e base funcional está no cerne do aparecimento das bibliotecas digitais assim como explica a complexidade do problema de engenharia em questão que se articula sob a forma de sistemas de interoperabilidade.

Seguidamente, apresentam-se a arquitectura Dienst e o modelo DELOS, que no todo ou parcialmente têm presidido à orientação criativa de muitas bibliotecas digitais. A arquitectura Dienst por ter sido uma das primeiras e o modelo DELOS por ser um dos mais recentes e continuar em revisão no sentido de ser melhorado (DL.org, 2009).

### 3.2.1 Arquitectura Dienst

Com origem no projecto DARPA – *Computer Science Technical Report* a arquitectura Dienst (Lagoze and Davis, 1995a; Lagoze et al., 1995b) nasceu com o intuito de providenciar uma biblioteca digital de relatórios técnicos das ciências da computação, tendo funcionado como base técnica na fundação NCSTRL – *Networked Computer Science Technical Reports Library* (Davis and Lagoze, 1996). A arquitectura desenvolve-se com base num conjunto de servidores distribuídos pela Internet facilitando o acesso a uma colecção de documentos multiforme, distribuídos e descentralizados.

Com o objectivo de armazenar conteúdos em múltiplos formatos (texto, imagens, vídeo, áudio, etc.) e disseminar os mesmos em múltiplas variações, a Dienst utiliza um

modelo de documentos desenvolvido para o efeito. Em detalhe, as suas características são:

- Atribuição de nomes globalmente únicos aos documentos, utilizando *handles* e num esquema URN, com nomes que consistem numa autoridade de nomeação única e uma *string* de identificação que também é única dentro dessa autoridade;
- Versões múltiplas dos documentos;
- Estruturação lógica dos documentos:
  - Múltiplos tipos de descrição de metadados que podem ser associados com o documento ou partes do mesmo (ex. capítulos, páginas, etc.);
  - Múltiplas representações de um documento, que são expressões ou formas alternativas de representação do conteúdo encapsulado no documento;
  - Estruturação hierárquica de cada vista do documento, como secções, capítulos e páginas.
- Múltiplos tipos de conteúdo, por exemplo tipos MIME, mecanismos agregadores, para permitir encapsular múltiplos objectos lógicos, e esquemas de compressão.

Para potenciar o uso deste modelo de documentos e permitir criar bibliotecas digitais distribuídas, a arquitectura é implementada sobre a noção de vários serviços: repositório, indexação, motor de busca (na Figura 3.3, identificados como QM - *Query Mediators*), serviços de nomeação (na Figura 3.3, identificados como NS - *Naming Services*) e interfaces com o utilizador para acesso e busca dos documentos.

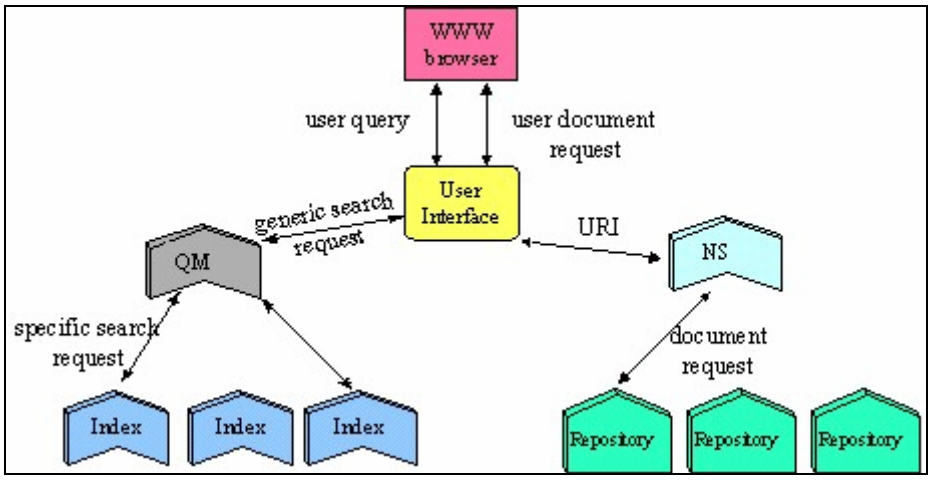



Figura 3.3 – Estrutura dos serviços Dienst (Cornell, 2000).

Implícita na interacção dos serviços está a sua capacidade individual de encaminhar os pedidos de busca através dos vários serviços. A arquitectura Dienst introduziu a noção de colecção de serviços que é assim responsável por providenciar a informação que permite a um conjunto de serviços interagir em conjunto. A Dienst define deste modo a colecção como um predicado dos serviços e recursos da biblioteca digital. A especificação de colecção implementada na Dienst passa por listar as organizações que fazem parte da colecção; identificar a localização dos servidores que indexam a informação por cada organização; e criar metainformação sobre cada um dos servidores de índice que ajude a encaminhar os pedidos de pesquisa.

Veja-se na Figura 3.4 o caso de uma busca, tendo em conta a colecção ciências da computação (CS). É iniciada uma pesquisa na interface com o utilizador, serviço UI, e o motor de busca, serviço QM, vai utilizar informação presente na colecção CS para encaminhar a busca no sentido dos índices, serviços I, mais apropriados.

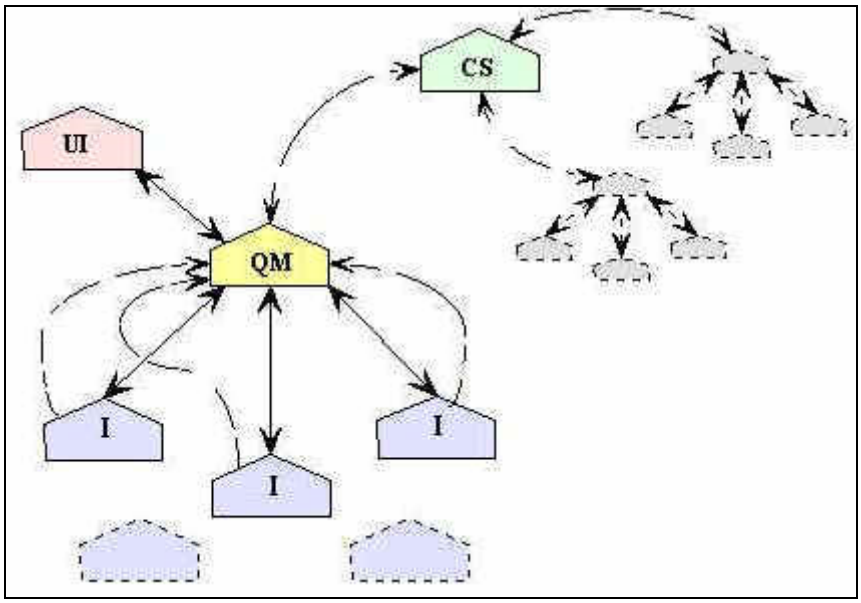



Figura 3.4 – Uso do serviço de colecções na arquitectura Dienst (Cornell, 2000).

Acerca da arquitectura Dienst, interessa reter e evidenciar o facto de os seus criadores a terem concebido como um conjunto de serviços independentes, cuja composição e interacção entre si instancia o sistema no seu todo e permite a criação de bibliotecas digitais de carácter distribuído.

### 3.2.2 Modelo DELOS

O modelo de referência DELOS para bibliotecas digitais (Candela et al., 2007) foi desenvolvido pela iniciativa DELOS – *A Network of Excellence on Digital Libraries*, já referenciada nesta dissertação, e tem por objectivo contribuir para a criação de fundações comuns, que permitam estabelecer uma melhor compreensão, comunicação e estimular a evolução futura do contexto complexo em que consiste o universo das bibliotecas digitais.

Comparativamente com a arquitectura Dienst, o modelo DELOS introduz uma mudança de paradigma no sentido em que deixa de apresentar uma perspectiva somente centrada nos conteúdos, organização e partilha de colecções de dados, para passar a centrar-se na pessoa com o objectivo de oferecer experiências estimulantes e personalizadas aos seus utilizadores. Assim, a mudança opera-se dos modos estáticos de acesso à informação para a facilitação da comunicação e colaboração entre os utilizadores. Outra mudança importante no paradigma, é a possibilidade de passar a considerar a biblioteca digital como uma eventual federação de bibliotecas digitais distribuídas e independentes.

Como tal, a visão do modelo DELOS é bastante mais abrangente e complexa e já não vê o problema apenas como uma Biblioteca Digital (BD), mas antes como um sistema ou universo de bibliotecas digitais.

Na Figura 3.5 encontra-se um esquema representando o modelo tripartido defendido pelo DELOS, com três níveis conceptuais do universo das bibliotecas digitais: o sistema de gestão das bibliotecas digitais (*Digital Library Management System*), o sistema das bibliotecas digitais (*Digital Library System*) e as bibliotecas digitais em si (*Digital Library*). Neste esquema presencia-se a riqueza de todo um sistema de bibliotecas digitais, que tem por meta potenciar a integração de sistemas, fazendo uso de tecnologias que facilitam a interoperabilidade, de forma a oferecer um acesso distribuído à informação em formatos heterogéneos.

Ao seu mais alto nível, o modelo apresenta-se devidamente conceptualizado, com conceitos como:

- Conteúdo – representa o ponto de entrada para todos os conceitos relacionados com o conteúdo que é gerido e disseminado pela biblioteca digital;
- Utilizador – funciona como matriz de conceitos, comunidades, perfis e identidades que representam aspectos dos utilizadores da BD;
- Funcionalidade – é a entrada para área das funções da BD;

- Arquitectura – está relacionada com os componentes de software, a gestão de nós e a forma como estão ligados e limitados;
- Qualidade – agrupa parâmetros qualitativos que caracterizam os comportamentos da BD em determinados contextos;
- Política – está relacionada com todos os conceitos, planos e procedimentos que governam as acções da BD, tais como gestão de colecções, preservação, direitos de acesso, etc.

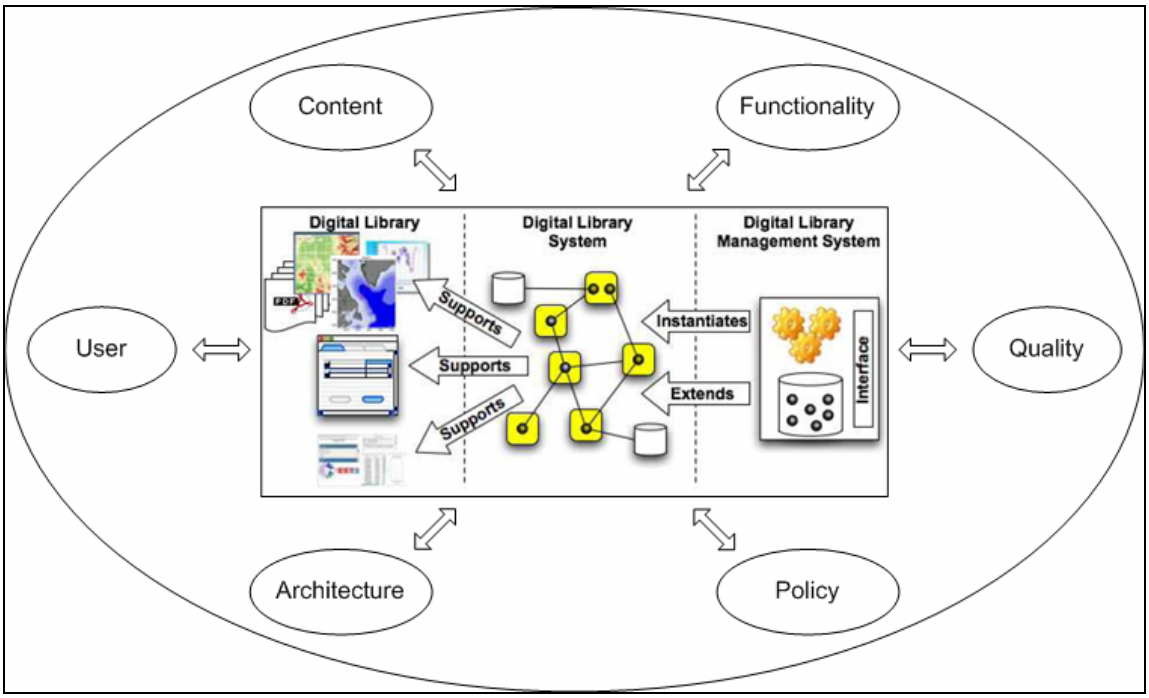



Figura 3.5 – Modelo de referência DELOS (Candela et al., 2007).

Finalmente e tendo em conta a relevância da interoperabilidade para este trabalho, atente-se no modo como o DELOS gere esta problemática.

Partindo do seu modelo de referência pode-se rapidamente comparar e identificar as diferenças e as similaridades e é daqui que se parte para uma análise dos problemas de interoperabilidade de modo a instituir um plano que os permita redimir.

Abordando o assunto a partir do seu modelo de referência, o DELOS identifica a questão da interoperabilidade como uma propriedade multidimensional, evitando a tradicional abordagem centrada meramente nos metadados e protocolos, e assume a necessidade de aplicar o princípio a todos os recursos dos diferentes domínios do

universo da Biblioteca digital, isto é: conteúdo, funcionalidade, utilizadores, qualidade, política e arquitectura.

Esta abordagem traz novas implicações a serem tomadas em conta pelo programador, que deste modo não pode mais preocupar-se apenas com a procura de cruzamentos entre os formatos de metadados, mas necessita, por exemplo, de mecanismos de medição da qualidade do conteúdo que assegurem os mesmo níveis de qualidade nos recursos participantes da BD.

Deste modo, o modelo DELOS procurou implementar vários conceitos tais como:

- "Resource <hasMetadata> Information Object" – permite capturar qualquer tipo de metadados para ajudar no suporte à interoperabilidade;
- "Resource <hasFormat> Resource Format" – permite capturar o Formato dos Recursos com o qual os Recursos sejam compatíveis. Por exemplo, para que BD A possa utilizar um objecto de informação BD B, BD A terá de ser capaz de lidar com o formato de informação do tipo desse objecto (ler, mostrar, etc.) ou, por outro lado, ser capaz de o converter para um formato com o qual consiga lidar;
- "Ontology" – é importante ao nível do Formato dos Recursos. As especificações dos formatos necessitam de ser preservados para que os objectos de informação mais antigos possam continuar a ser interpretados. Do mesmo modo as diferentes versões das ontologias precisam de ser preservadas;
- "Resource <associatedWith> Resource" – é responsável por capturar o contexto a partir do qual um determinado Recurso foi originado. Esta componente funciona como uma base de conhecimento importante para a correcta interpretação do significado do Recurso.

Assim como várias funções:

- "Transform" – é uma especialização do "Process Function of Manage Information Object". Esta função inclui propriedades como a conversão de formatos, a extracção de informação, a tradução automática e técnicas para sumariar. A propriedade de conversão inclui a conversão para diferentes codificações, tais como converter um texto de formato PDF para formato Word, ou uma imagem para um outro formato ou compressão;
- "Import Collection" – aparece como uma especialização da função "Manage Collection" e suporta a selecção de informação a partir de fontes externas para serem utilizadas pela BD como conteúdo ou recursos de metadados;

- "Export Collection" – mais uma especialização da função "Manage Collection", que suporta a DB na sua totalidade ou em partes, de modo a permitir a criação de *mirrors* e/ou cópias de segurança. Para além disso permite disponibilizar os Objectos de Informação, especialmente os metadados, a outros sistemas de colecta de informação;
- "Compare" – uma especialização da função "Analyze Function" e que pode ser utilizada para averiguar se duas instâncias de um Objecto de Informação são iguais.

Desde a sua criação, sujeito a contínuos refinamentos, o modelo DELOS apresenta-se assim como um modelo extremamente avançado, tentando abranger a grande complexidade que rodeia o domínio multidisciplinar das bibliotecas digitais.

## 3.3 Plataformas

Existem actualmente múltiplas e diversas plataformas de software disponíveis para a criação e gestão de repositórios digitais, que podem ser utilizadas no suporte às bibliotecas digitais. No âmbito do código aberto, são exemplos de referência: a Greenstone (NZDLP, 2007), a Fedora (Staples e tal., 2003; Fedora, 2009), a DSpace (DSpace, 2009a), a CDS Invenio (CDSsc, 2005) e a EPrints (EPrints, 2009). No âmbito dos sistemas proprietários, podem-se referir: a MetaLib (Ex_Libris, 2008), a CONTENTdm (OCLC, 2009b) e a Veridian (DLconsulting, 2009).

É difícil propor uma plataforma específica como sendo a mais indicada para todos os cenários. Cada plataforma possui as suas próprias vantagens e desvantagens, como se constata em alguns estudos comparativos de repositórios digitais para a criação de bibliotecas digitais (NRGL, 2009; Pyrounakis e Nikolaidou, 2009; Goutam e Dibyendu, 2010).

Para a construção de uma biblioteca digital, em geral, uma organização tem de decidir qual das plataformas de suporte deve usar por forma a conseguir hospedar as suas colecções de forma eficiente. As necessidades para cada organização variam com o número de colecções, o tipo de objectos, a natureza do material, a frequência das actualizações, a distribuição dos conteúdos e os limites temporais para o seu desenvolvimento. Desta forma, é óbvio que a escolha varie conforme o cenário presente em cada organização e não seja possível dizer qual das plataformas é a melhor, em sentido absoluto.

Nesta secção passam-se em análise duas plataformas, a DSpace e a MetaLib, em representação dos sistemas de código aberto e dos sistemas proprietários, respectivamente. A DSpace, porque é uma das mais utilizadas actualmente. O seu site web conta, até ao momento, já cerca de um milhar de registos de instâncias de repositórios DSpace activas (DSpace, 2009c). A MetaLib, porque possui uma abordagem diferente aos repositórios digitais. Ao invés da criação de repositórios de raiz, permite o acesso e gestão de múltiplos repositórios digitais heterogéneos, já existentes, através de um único sistema, capaz de oferecer aos seus utilizadores a vantagem do acesso e manipulação, de forma integrada, da informação provinda de múltiplas fontes.

### 3.3.1 DSpace

A plataforma DSpace (DSpace, 2009a) é um sistema de software de código aberto, concebido para realizar a gestão de elementos digitais, sendo principalmente usado como plataforma de repositórios institucionais.

Esta plataforma foi tornada pública em 2002 depois de um esforço de investigação conjunta entre o MIT e a HP Labs. Em 2004, foi criada a DSpace Federation com o intuito de assegurar o futuro da plataforma e do DSpace Committers Group, constituído por cinco instituições – MIT, HP Labs, OCLC, Universidade de Cambridge e Universidade de Edimburgo – às quais se juntaram mais tarde a Universidade Nacional da Austrália e a Universidade do Texas A&M. Em 2005, foi lançada a versão 1.3. Entretanto em 2007, novamente o MIT e a HP Labs anunciaram (HP, 2007) a formação da DSpace Foundation, uma organização sem fins lucrativos com o objectivo de providenciar liderança e suporte à comunidade da DSpace. Já em 2008 foi lançado a DSpace 1.5. e entretanto no decorrer do ano de 2009, a DSpace Foundation fundiu-se com a Fedora Commons para criar a nova organização DuraSpace (DuraSpace, 2009), tendo sido efectuadas algumas melhorias à plataforma e lançada a mais recente versão 1.5.2.

O diagrama da plataforma DSpace (Figura 3.6) evidencia o percurso do material digital desde que este entra no sistema até chegar ao utilizador final. Este material digital é tratado e armazenado no sistema segundo um modelo de dados próprio.

À entrada, o material digital é captado utilizando um módulo de submissão processual. A organização do espaço é realizado segundo comunidades, que podem conter sub-comunidades, reflectindo desta forma uma clara analogia com a estrutura de uma universidade que se subdivide em departamentos, centros de investigação e laboratórios. As “comunidades” por sua vez contêm “colecções” – grupos de conteúdos relacionados.

As colecções podem ainda pertencer a mais do que uma comunidade. Aos elementos constituintes das colecções é dado o nome de "item" que funciona como unidade básica de armazenamento. Por sua vez os "itens" são constituídos por pacotes de *bitstreams*, ou seja ficheiros.

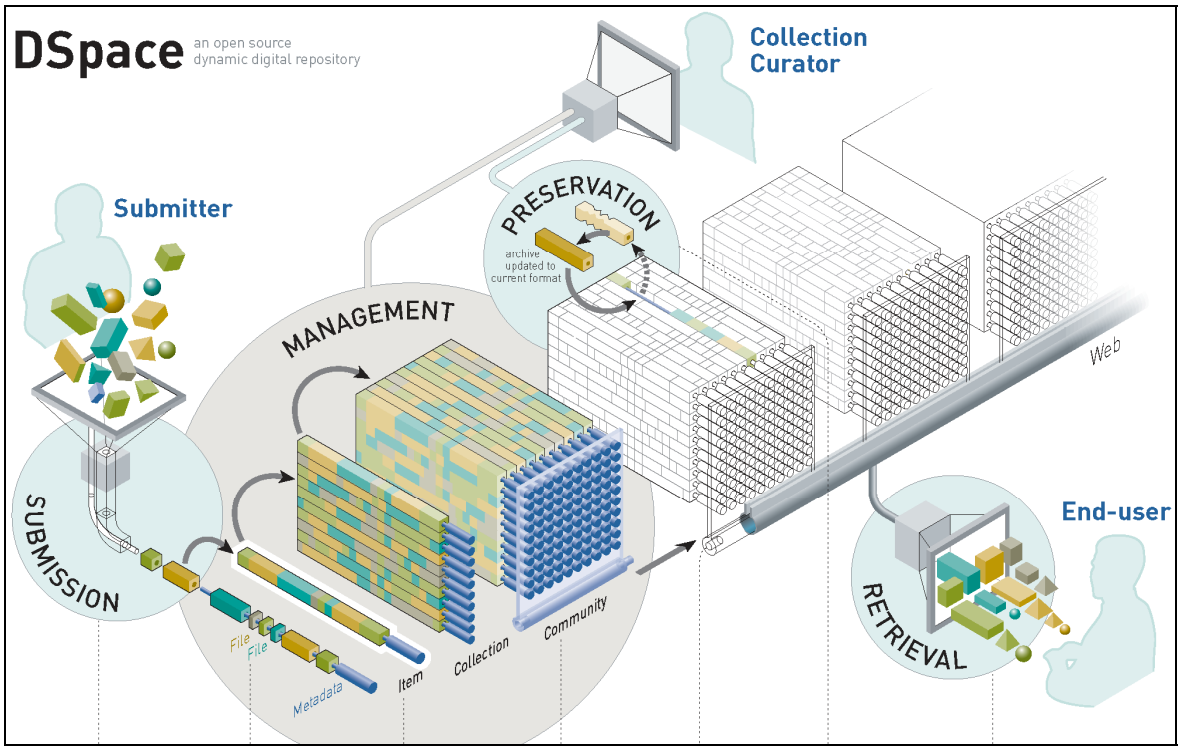



Figura 3.6 – Diagrama da plataforma DSpace (DSpace, 2009b).

Em relação aos metadados, estes são divididos em três categorias: descritivos, administrativos e estruturais. Sendo os descritivos definidos com recurso a um esquema de Dublin Core baseado no *Library Application Profile* (DCMI, 2004), com amplas possibilidades de configuração no que toca à descrição dos "itens". A categoria dos administrativos inclui metadados de preservação, proveniência e políticas de autorização. Quanto aos estruturais, estes funcionam de modo simples e detêm informação acerca da forma como os "itens" são apresentados, ou os *bitstreams* no interior de um "item", e a relação entre cada um dos elementos desse mesmo "item".

Toda a informação submetida ao repositório é, depois de passar pelo sistema de catalogação e gestão, arquivada e preservada em blocos de longo termo. Finalmente encontra-se em posição de ser visualizada pelos utilizadores finais, que através de uma

interface baseada na web e de um sistema de pesquisa e recolha de informação, fazem a sua requisição.

### 3.3.2 MetaLib

A plataforma MetaLib (Ex_Libris, 2008) é um sistema proprietário, desenvolvido e mantido por uma empresa, a Ex Libris. Define-se como um portal de acesso a colecções de bibliotecas, ou seja, fornece um ambiente integrado capaz de gerir recursos electrónicos em bases de dados distribuídas e heterogéneas.

O modo de acesso à plataforma MetaLib pode ser realizado de duas formas:

- através de acessos directos a ligações (*links*) existentes na interface, que funcionam como "Information Gateway";
- ou através do motor de busca, a interface "MetaLib Search", acedendo assim ao que a MetaLib chama de "Universal Gateway" (Lewis, 2002).

Em termos de especificidade funcional, a MetaLib suporta grande parte das normas em vigor na indústria e nas bibliotecas:

- compatível com o formato Unicode (Unicode, 2009);
- segue o guia *Web Content Accessibility Guidelines 1.0 (Level A)*, da W3C (W3C, 1999a), para aumentar a acessibilidade de pessoas com deficiências;
- para esquema dos registos, utiliza principalmente MARC (LoC, 2006) e Dublin Core (DCMI, 2008);
- para formato dos dados, utiliza XML (W3C, 2008) e HTML (W3C, 1999b);
- e ao nível dos protocolos de pesquisa, utiliza o Z39.50 (ZIG, 2003), o SRU/SRW (LoC, 2009c; LoC, 2009d), o HTTP (NWG, 1999) e o NISO MXG (NISO, 2009).

## 3.4 Protocolos de Pesquisa e Recolha

No âmbito das bibliotecas digitais, a interface de pesquisa é um dos tópicos mais importantes para o utilizador. A presença de uma interface de pesquisa num sistema de informação, seja ele de que tipo for, altera por completo o paradigma da procura de informação por parte de um utilizador, pois este deixa de estar à mercê de uma listagem de apontadores, que foram criados previamente, segundo critérios que lhe são completamente alheios, para passar a deter o controlo sobre que informação pertencente ao sistema pretende consultar.

Contudo, a pesquisa a sistemas de informação, como bases de dados por exemplo, com recurso a motores de pesquisa tradicionais (Google, Bing, etc.), oferece imensas resistências. Por isso, estes sistemas são muitas vezes classificados de web profunda (*deep web*). Os factores cruciais para estes sistemas de informação se tornarem invisíveis aos "olhos" da pesquisa passam pelo facto de se tratar de informação que requer conhecimento prévio acerca do seu modo particular de acesso, ou seja, são sistemas que funcionam como uma "web" paralela, na qual é necessária uma chave para o motor de pesquisa poder entrar.

Fica então claro porque é que a pesquisa a sistemas de informação complexos, como as bibliotecas digitais, tem de obrigatoriamente recorrer a mecanismos que "conheçam" a especificidade desses sistemas, ou seja, é necessária utilização de protocolos de pesquisa e recolha que estejam conformes à funcionalidade de pesquisa oferecida pelo sistema e conformes ao formato e semântica dos dados a pesquisar.

São inúmeros os protocolos de pesquisa e recolha que podem ser adoptados pelas bibliotecas digitais, com o objectivo de fornecerem acesso aos seus repositórios de informação. Com o intuito de apresentar alguns desses protocolos, são descritos, muito sumariamente, o protocolo Z39.50 e os protocolos OAI. O primeiro é tido, pela comunidade de implementadores de sistemas, como um protocolo "antigo e complexo"; os segundos como protocolos "modernos e mais simples".

### 3.4.1 O Protocolo Z39.50

O protocolo Z39.50 (ZIG, 2003) é um protocolo bastante maduro com inícios de desenvolvimento nos anos 70 e conta com várias versões e actualizações desde então. Este protocolo obedece às normas internacionais, ISO 23950: "*Information Retrieval (Z39.50): Application Service Definition and Protocol Specification*" e ANSI/NISO Z39.50, reguladas pela Biblioteca do Congresso (*The Library of Congress*) dos EUA (LoC, 2009b).

Uma das principais e mais importantes características deste protocolo, reside na sintaxe de pesquisa. Este protocolo utiliza um conjunto de descritores de pesquisa que é abstraído da estrutura de dados, ou seja, a forma como é mapeada a informação no processo de busca está baseada no servidor. Isto permite que o Z39.50 realize buscas sem "conhecer" os formatos presentes nas bases de dados, o que torna a busca possível em qualquer cenário, embora correndo o risco de por vezes os resultados apresentarem variações provenientes dos diferentes modos de catalogação das bases de dados.

Sendo o Z39.50 uma tecnologia desenvolvida nos anos 70 e sem preocupações web, existem actualmente vários grupos a trabalhar a sua evolução para uma chamada "nova geração", utilizando várias estratégias. Destas a mais importante tem sido perseguida pelo protocolo SRU - *Search/Retrieval* via URL (LoC, 2009c) que opta por usar como protocolo de transporte o HTTP, utilizando directamente os seus métodos GET ou POST. Em alternativa, pode ser usado o protocolo SOAP (W3C, 2007a), sobre HTTP, o que corresponde formalmente ao protocolo SRW (LoC, 2009d), oferecendo assim uma interface de comunicação baseada em *web services* (W3C, 2004b). A utilização destes protocolos de transporte não modifica em nada o modo de pesquisa que utiliza os mesmos serviços funcionais e os mesmos conjuntos de descritores de pesquisa. Apenas neste caso, os resultados devolvidos são sempre codificados no formato XML.

### 3.4.2 Os Protocolos OAI

A iniciativa para os arquivos abertos, OAI - *The Open Archives Initiative*, é um dos mais importantes esforços contributivos para a resolução dos problemas de interoperabilidade técnica entre os arquivos distribuídos (Lagoze and Van de Sompel, 2001).

O objectivo da OAI é facilitar a descoberta de conteúdos em arquivos distribuídos e difere de outras iniciativas, como a do protocolo Z39.50, através da enfatização numa implementação mais fácil, simples e limitada. Presentemente, a OAI propõe dois protocolos para atingir este objectivo: o OAI-PMH e o OAI-ORE.

O protocolo OAI-PMH - *Open Archives Initiative Protocol for Metadata Harvesting* (OAI, 2008a) é utilizado para recolher e coligir registos de metadados a partir de múltiplos arquivos distribuídos e define dois papéis funcionais principais nesse processo: o de provedor de dados (repositórios) e o de provedor de serviços.

Os provedores de serviços extraem os registos de metadados dos repositórios através do protocolo para um arquivo local e depois oferecem serviços de valor acrescentado sobre a informação recolhida. Esses serviços podem ser implementados na forma de sistemas ou motores de pesquisa, sistemas referenciais e/ou sistemas de revisão entre pares, através dos diversos repositórios distribuídos.

Os pedidos de pesquisa formulados pelos provedores de serviços aos repositórios, são expressos através dos seis verbos constantes na especificação do protocolo: *GetRecord*, *Identify*, *ListIdentifier*, *ListMetadataFormats*, *ListRecords* e *ListSets*. O transporte destes pedidos é efectuado pelo protocolo HTTP, utilizando os seus métodos

GET ou POST. As respostas dos repositórios são expressas através das respostas HTTP, codificadas no formato XML.

Embora os repositórios possam oferecer mais que um formato para os registos de metadados, estes têm de disponibilizar obrigatoriamente o formato Dublin Core simples (DCMI, 2008).

O protocolo OAI-ORE - *Open Archives Initiative Object Reuse and Exchange* (OAI, 2008b) define normas para a descrição e transferência de recursos web agregados. Estes agregados são por vezes chamados objectos digitais compostos e podem combinar recursos distribuídos com diferentes tipos de conteúdo, como texto, imagens, vídeos, dados, etc.

Estas normas lançam as fundações para expor o conteúdo rico de recursos agregados a aplicações e serviços que suportem a sua visualização, autoria, reutilização, transferência, armazenamento e preservação, assim como aumentar e melhorar o acesso aos agregados utilizados pelas pessoas na sua interacção diária com a web. Estes agregados incluem: documentos compostos por conjuntos de páginas web, documentos com múltiplos formatos em repositórios institucionais, conjuntos de dados académicos ou colecções de música e fotografia.

As normas OAI-ORE utilizam conceitos oriundos da Arquitectura Web (W3C, 2004a) e de esforços relacionadas com a web semântica (W3C, 2009), Linked Data (LData, 2009) e Atom Syndication (NWG, 2005). Como resultado, integram-se com a emergente Web 2.0 e com a futura evolução da rede de informação.

Por forma a que um agregado de recursos web seja referido sem ambiguidade, o modelo de dados do OAI-ORE introduz o conceito de agregado (*aggregation*), um novo recurso, que consiste num conjunto ou colecção de recursos. O agregado é uma construção conceptual e por isso qualifica-se como um dos recursos da web semântica que não possui uma representação. A complexidade de um agregado pode assumir grandes proporções, visto este poder agregar outros agregados. Um exemplo extremo é o de considerar todo o repositório como um agregado, que por sua vez detém agregados menores.

Para a descrição dos agregados, nomeadamente os recursos que deles fazem parte, o modelo introduz ainda um recurso chamado mapa de recursos (*resource map*), que só pode estar associado a um agregado.

Tanto o agregado como o mapa de recursos possuem URIs (W3C, 2001) associados, para que o protocolo funcione sobre a web.

## 3.5 Normas de Metadados

Como se verificou no ponto anterior, as normas de metadados são estruturas essenciais na codificação dos sistemas de pesquisa das bibliotecas digitais, ou seja, funcionam como estruturas de representação e comunicação de dados entre máquinas.

Apresentam-se seguidamente duas das normas mais utilizadas actualmente para a codificação de metadados: o MARC e o Dublin Core.

### 3.5.1 MARC

O formato MARC - *Machine-Readable Cataloging Record* (LoC, 2006) consiste numa norma de metadados que envolve três elementos: a estrutura do registo, a designação do conteúdo, e o conteúdo do registo.

A estrutura dos registos MARC passa pela implementação de outras normas internacionais, tais como o ANSI Z39.2 - *Information Interchange Format* (NISO, 1994) ou o ISO 2709 - *Format for Information Exchange* (ISO, 2008).

O elemento de designação de conteúdos refere-se aos códigos e convenções estabelecidos para identificar e caracterizar os dados presentes num registo.

Finalmente os conteúdos do registo são definidos pelas normas externas como, por exemplo: o AACR2 - *Anglo-American Cataloguing Rules* (AACR, 2006) ou o LCSH - *Library of Congress Subject Headings* (LoC, 2009e).

A informação que um registo MARC carrega é definida por cinco tipos de dados:

- bibliográficos – contêm especificações para a codificação de elementos necessários para descrever, recolher e controlar vários tipos de materiais bibliográficos, tais como: livros, séries, ficheiros de computador, mapas, música, materiais visuais, entre outros;
- de propriedade – possuem informação capaz de designar a propriedade e a localização para todas as formas de materiais;
- de autoridade – são responsáveis por identificar e controlar os conteúdos bibliográficos sujeitos a controlos de autorizados;
- de classificação – apresentam especificações para a codificação de dados relacionados com números de classificação e as descrições com estes associados, servindo no sentido de ajudar à manutenção e desenvolvimento de esquemas de classificação;

- de informação comunitária – apresentam informação sobre eventos, programas, serviços, entre outros, de forma a serem integrados em catálogos de acesso público.

### 3.5.2 Dublin Core

O Dublin Core (DCMI, 2008; Hillman, 2005) é, à semelhança do MARC, uma norma de metadados que tem por objectivo descrever um alargado número de recursos em rede.

Esta norma está dividida em dois níveis: Simples e Qualificado. O nível simples é constituído por quinze elementos: *Title*, *Subject*, *Description*, *Type*, *Source*, *Relation*, *Coverage*, *Creator*, *Publisher*, *Contributor*, *Rights*, *Date*, *Format*, *Identifier*, e *Language*. O nível qualificado adiciona mais três elementos: *Audience*, *Provenance*, *RightsHolder*. Este nível define também subelementos qualificadores para os elementos.

Cada um dos elementos é opcional e pode mesmo ser repetido. No nível qualificado, a maior parte dos elementos possui um conjunto limitado de qualificadores, ou seja, atributos que podem ser utilizados para refinar o significado de cada elemento. A DCMI - *Dublin Core Metadata Initiative* estabeleceu normas para refinar os elementos e encoraja o uso de esquemas de vocabulário e codificação. Para proceder a tal, existem três princípios no Dublin Core: *One-to-One Principle*; *Dumb-down Principle*; e *Appropriate Values*.

O primeiro princípio, *One-to-One*, parte da questão genérica em que os metadados Dublin Core descrevem uma manifestação ou versão de um recurso, em vez de assumir que as várias manifestações servem apenas o recurso. Ou seja, as cópias são vistas como objectos independentes e não como meras extensões do original. A relação entre o original e a cópia aparece declarada nos metadados.

O segundo princípio, *Dumb-down*, serve na qualificação das propriedades. Este princípio define que o cliente pode ignorar qualquer qualificador e utilizar apenas o elemento, como se este fosse não qualificado. Apesar da perda de especificidade, o valor restante do elemento continua a ser de uma forma geral correcto e ao mesmo tempo útil para a descoberta. A qualificação é assim utilizada apenas para refinar e não estender o âmbito semântico da propriedade.

No último princípio, *Appropriate Values*, as melhores práticas para cada elemento ou qualificador podem variar com o contexto, mas em geral o implementador não pode prever que o intérprete dos metadados seja sempre uma máquina. Esta questão pode

impor certas limitações aos metadados, mas o requisito de utilidade para a descoberta deve ser sempre mantido.

## 3.6 Revisão

Neste capítulo, passou-se em revista as tecnologias mais utilizadas na área das Bibliotecas digitais. A principal preocupação centrou-se sobre os modelos, arquitecturas, plataformas, protocolos de pesquisa e recolha e as normas de metadados. Estes pontos não encerram todas as necessidades tecnológicas necessárias à execução de um universo de bibliotecas digitais, mas representam contudo, no entender do autor desta dissertação, os seus pontos-chave e o conhecimento necessário à implementação do sistema proposto por este projecto de doutoramento.

# Capítulo 4

# Plataforma de *Middleware* de Suporte a Bibliotecas Digitais Distribuídas

## 4.1 Introdução

Desde o início da investigação no domínio das bibliotecas digitais, foram já desenvolvidos múltiplos e diversificados sistemas para a implementação de arquivos e bibliotecas digitais. A grande maioria desses sistemas, foram desenvolvidos com o objectivo de serem utilizados em cenários específicos cobrindo áreas e domínios específicos da informação. A abertura desses sistemas à diversificação da informação e o seu envolvimento com outros sistemas, permaneceu contudo algo limitado.

Este cenário levou, no âmbito deste projecto de doutoramento, à concepção de uma plataforma de *middleware*, possuidora de suficiente flexibilidade, para permitir a acomodação, a instanciação e a expansão de bibliotecas digitais de índole específica ou genérica. Contudo, a sua característica mais importante será, porventura, a sua capacidade de integração de sistemas, de metadados e de conteúdos (informação multimédia), provenientes de múltiplas bibliotecas ecléticas. Esta é a característica chave deste trabalho, ainda não vislumbrada, na sua plenitude, nos muitos sistemas de bibliotecas digitais desenvolvidos até hoje.

Para a concepção e posterior desenvolvimento desta plataforma foram elaborados vários pontos de que se dão conta de seguida:

- um conjunto de requisitos de utilizador e de sistema;
- um modelo de abstracção genérico para a materialização de ideias e conceitos que se pretendiam ver presentes na plataforma;
- uma arquitectura específica, fazendo referência a tecnologias específicas, para a consubstanciação do modelo;
- a definição das principais interfaces funcionais e dos modelos de dados manipulados por essas interfaces.

A plataforma de *middleware* concebida a partir de todos estes elementos pretende conduzir ao desenvolvimento de sistemas, o menos limitadores e o mais abertos possíveis, com um espectro de aplicação o mais largo possível.

## 4.2 Requisitos

Com base em múltiplas considerações tecidas por diversos autores (Lagoze et al., 1995b; Candela et al., 2007) no desenvolvimento de modelos e arquitecturas para bibliotecas digitais e após longa reflexão sobre as potenciais funcionalidades a acomodar na plataforma aqui apresentada, foi elaborado um conjunto de requisitos para nortearem a sua concepção. Estes requisitos dividem-se essencialmente em 2 grandes grupos: os requisitos de utilizador e os requisitos de sistema.

### 4.2.1 Requisitos de Utilizador

Os requisitos de utilizador estão essencialmente relacionados com as funcionalidades que o utilizador final poderá ver presentes no sistema. Estes requisitos passam pela possibilidade de:

- pesquisar informação em bibliotecas digitais múltiplas, diversas e distribuídas, a partir de um único ponto centralizado;
- possuir um maior grau de controlo sobre o desenrolar das pesquisas, podendo por exemplo indicar as fontes de informação a pesquisar, o número máximo de itens informativos pretendidos, o tempo máximo de espera pela conclusão da pesquisa, etc., para além da própria pesquisa;
- visualizar a informação recolhida em diferentes perspectivas: num conjunto único e indiferenciado ou em diferentes conjuntos, conformes às fontes da informação;

- aceder a toda a informação recolhida, desde simples referências a conteúdos multimédia, dentro do mesmo contexto do sistema.

### 4.2.2 Requisitos de Sistema

Os requisitos de sistema colocam a tónica sobre a existência de mecanismos internos no sistema que contribuem para que esta apresente determinadas características, como sejam por exemplo: maior fiabilidade, robustez, flexibilidade, etc.

No caso específico desta plataforma de *middleware*, pretende-se que esta apresente mecanismos que permitam:

- a interligação e interoperabilidade efectiva entre sistemas heterogéneos;
- a escalabilidade e fiabilidade do sistema;
- a implementação do paralelismo no acesso de múltiplos sistemas;
- o registo das características e da localização dos sistemas a aceder;
- a integração de modelos de metadados diferenciados;
- a identificação e eliminação de referências duplicadas.

## 4.3 O Modelo de Abstracção

Tendo por base alguns dos requisitos apresentados anteriormente e a existência de ideias e conceitos que se pretendiam ver incorporados no sistema, foi concebido um modelo de abstracção genérico que permitisse a materialização dos conceitos e a posterior concepção de arquitecturas. Este modelo encontra-se representado na Figura 4.1.

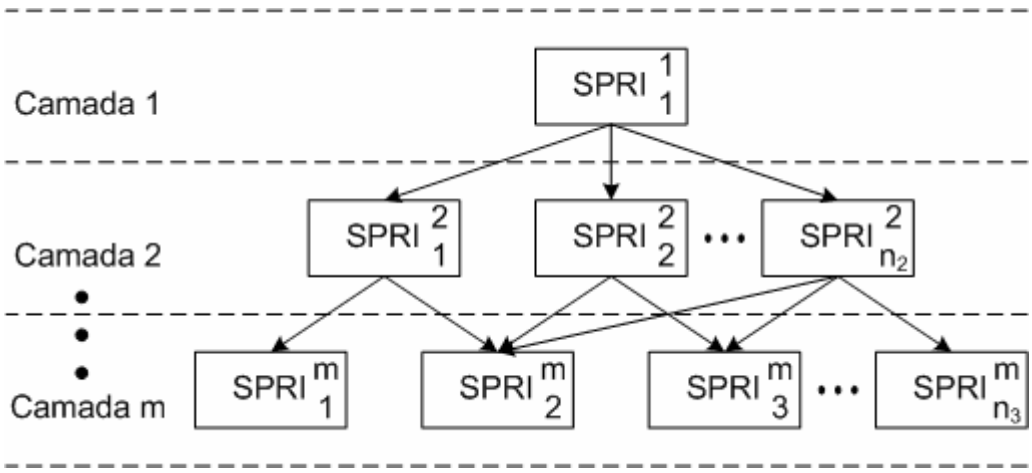



Figura 4.1 – Modelo de abstracção genérico para a plataforma de *middleware*.

O modelo de abstracção aqui apresentado é formulado como um conjunto de elementos funcionais, distribuídos por múltiplas camadas de abstracção, que têm por objectivo principal oferecer serviços de pesquisa e recolha de informação. Deste modo, foi dado a estes elementos o nome de SPRI - Serviço de Pesquisa e Recolha de Informação.

O SPRI é o elemento base do modelo e pode assumir um, vários ou todos os seguintes papéis: cliente, servidor ou repositório de informação. Em qualquer dos papéis, este elemento funcional tem de possuir sempre, de forma mais ou menos acentuada, as funcionalidades relacionadas com a procura e a recolha de informação.

Como explícito no modelo, um SPRI é um cliente para os SPRIs da camada abaixo e é um servidor ou repositório de informação para os da camada acima.

O número de SPRIs não é limitado, assim como o não é o número de camadas de abstracção. Do ponto de vista da escalabilidade, este modelo não apresenta limites teóricos, permitindo o crescimento da plataforma até limites fisicamente possíveis.

### 4.3.1 Conceitos Fundamentais

Neste modelo são utilizados alguns conceitos fundamentais que vão ao encontro dos requisitos especificados inicialmente e permitem a simplificação do próprio modelo. Esses conceitos são:

- a distribuição;
- o paralelismo;
- e a recursividade.

Este modelo fomenta a distribuição dos seus elementos funcionais e, como é visível na Figura 4.1, encontram-se distribuídos através das diferentes camadas e mesmo dentro da mesma camada. Este facto leva à implementação de sistemas escaláveis e consequentemente mais fiáveis, se adoptadas as estratégias de redundância adequadas.

É um modelo que insere o processamento paralelo como solução para a pesquisa, recolha e tratamento da informação, proveniente de múltiplas fontes. A interacção entres os vários elementos funcionais é feita de forma simultânea e paralela.

O conceito de recursividade aparece neste modelo como um meio de simplificar ao máximo a modelação da funcionalidade do sistema. Todas as camadas de abstracção do modelo são pautadas pelo mesmo comportamento funcional, ou seja, todos os elementos funcionais possuem genericamente a mesma funcionalidade, não só aqueles que se

encontram numa mesma camada, mas também aqueles que se situam em camadas diferentes. O que redunda numa reutilização conceptual da funcionalidade.

### 4.3.2 As Camadas de Abstracção

As camadas de abstracção, presentes neste modelo, são camadas conceptuais traduzidas pela necessidade que os elementos funcionais possam ter, ou não, de recorrerem a outros elementos para satisfazer a suas necessidades de obtenção de informação.

Quando um elemento funcional, a agir como um servidor de informação, não detém, ele próprio, a informação procurada, então este tem de agir também como cliente de outro elemento, na procura dessa informação. É este procedimento que gera, em si mesmo, o aparecimento das diversas camadas do modelo.

Assim, pode dizer-se que as camadas de abstracção neste modelo reflectem apenas a distância hierárquica entre os elementos que procuram informação e aqueles que, sendo fontes de informação, a podem disponibilizar.

Na última camada, encontram-se apenas elementos que são repositórios ou fontes originais de informação e que por isso não necessitam de consultar outros elementos, não gerando mais camadas. Contudo, não deve ser assumido que as fontes originais de informação se encontram todas na última camada. Estas podem encontrar-se igualmente em camadas intermédias.

### 4.3.3 Os Elementos Funcionais

Os elementos funcionais deste modelo – SPRIs – são elementos que têm por missão a procura e recolha de informação, podendo assumir vários papéis, como referido anteriormente. Com vista à compreensão da funcionalidade destes elementos, encontra-se na Figura 4.2 a representação do seu modelo genérico interno.

Um elemento funcional, SPRI, é genericamente composto por diversos subelementos que concorrem na sua execução para cumprir a tarefa de pesquisa e recolha de informação.

O subelemento, de nome Front-End, é aquele que oferece uma interface para interacção com o cliente do SPRI, seja recebendo pedidos de pesquisa ou de informação, seja devolvendo os resultados desses pedidos.

O subelemento, de nome Monitor, tem a seu cargo a instanciação dos subelementos Cliente e a monitorização do progresso das suas actividades.

Os subelementos Cliente são o meio pelo qual o SPRI interage com outros SPRIs com o intuito de proceder a pedidos de informação nesses elementos.

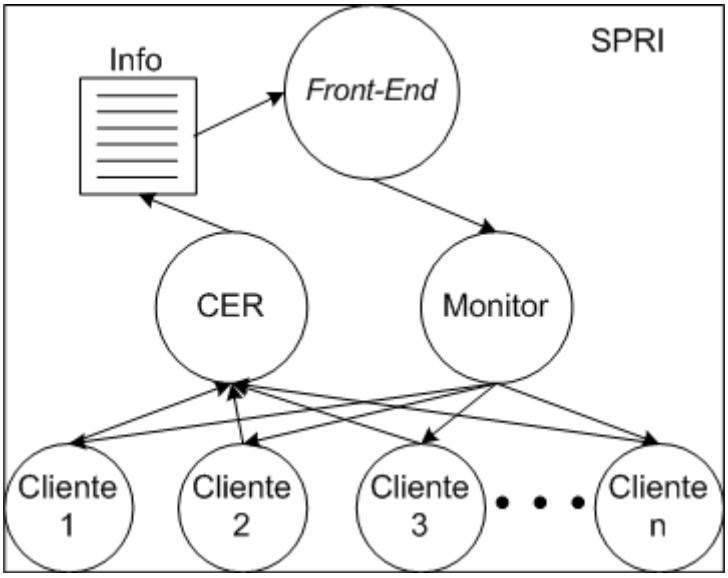



Figura 4.2 – Modelo genérico do SPRI.

Toda a actividade exercida por estes subelementos é desenvolvida em plena simultaneidade. Aqui se encontra, mais uma vez, o conceito de paralelismo na execução.

Da actividade dos subelementos Cliente surge como resultado um conjunto de informações que é enviado directamente ao subelemento CER - Conversor e Eliminador de Réplicas e tem por missão converter toda a informação recebida para um formato comum e posteriormente eliminar aquela que se encontra duplicada. À medida que estas actividades, de recolha e processamento de informação, se vão desenrolando, vai sendo estabelecido um depósito temporário de informação, o qual é utilizado pelo subelemento Front-End para satisfazer o pedido inicial de informação, assim como subsequentes pedidos relacionados com o pedido inicial.

Quando um SPRI assume o papel exclusivo de fonte de informação, poderão não estar presentes todos estes subelementos, uma vez que não há necessidade de procura de informação noutros elementos.

## 4.4 A Arquitectura

Com vista à consubstanciação da plataforma de *middleware*, baseada no modelo de abstracção anterior, foi concebida uma arquitectura que atribui papéis específicos aos

elementos funcionais do modelo – SPRIs – e emprega tecnologias específicas para ir ao encontro dos requisitos iniciais. Na Figura 4.3, encontra-se a representação dessa arquitectura.

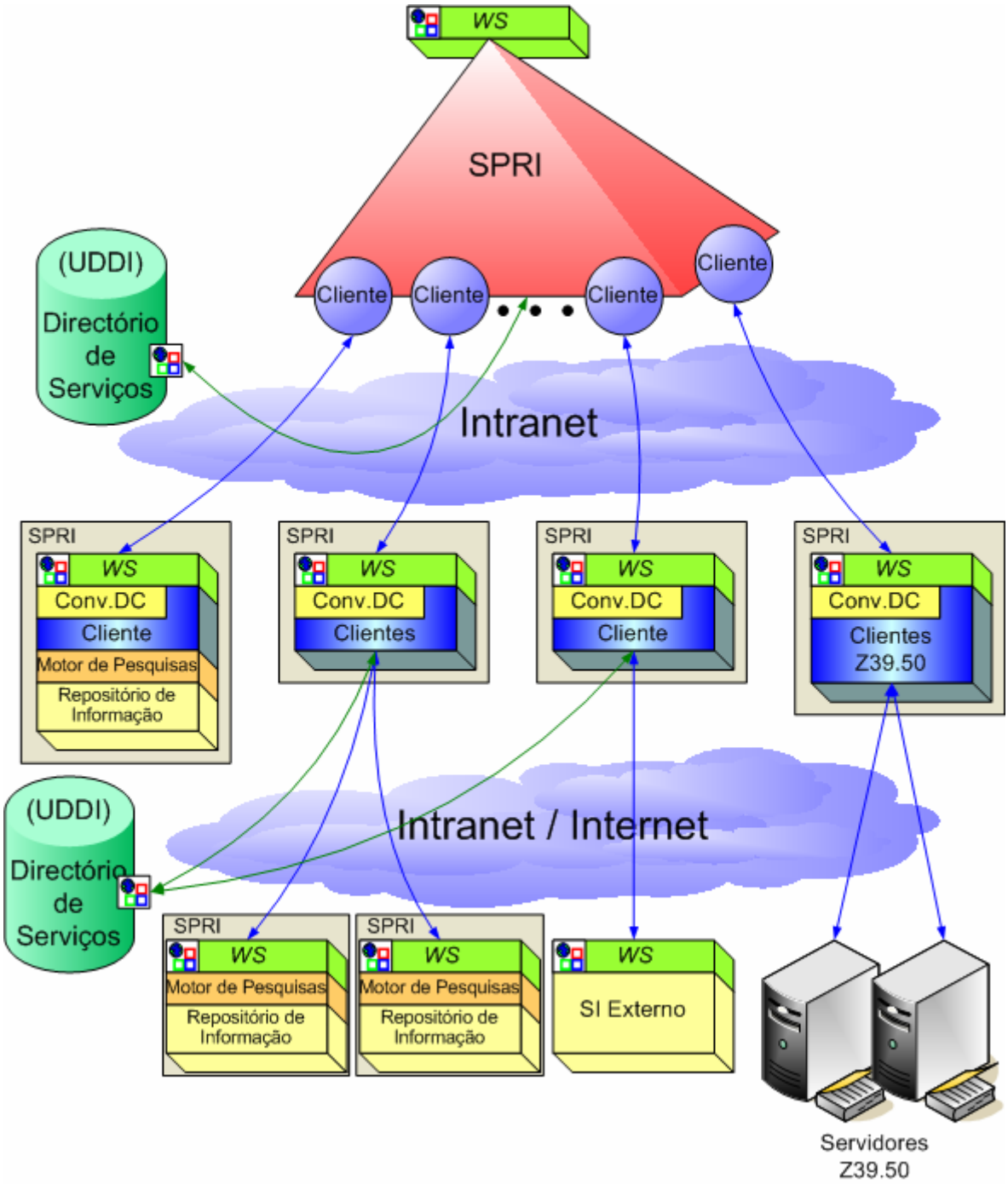



Figura 4.3 – Arquitectura para a plataforma de *middleware*.

Esta arquitectura apresenta uma estratificação em três camadas, aparentemente diferenciadas funcionalmente, e os seus elementos funcionais encontram-se totalmente distribuídos, numa intranet e na Internet. A aparente divisão funcional das camadas advém apenas da selectiva adjudicação dos diferentes papéis possíveis aos vários elementos funcionais, pois a arquitectura não pretende contrariar o modelo que lhe deu origem.

### 4.4.1 Os Papéis dos Elementos Funcionais nas Camadas de Abstracção

No topo da arquitectura, ocupando a primeira camada de abstracção, situa-se o SPRI responsável pelo atendimento de pedidos de pesquisa e de informação efectuados à plataforma. É neste caso o elemento funcional de entrada no sistema. Contudo, e com base no modelo de abstracção descrito anteriormente, este SPRI de topo poderia perfeitamente ser integrado num sistema maior, numa camada intermédia, fazendo com que nessa situação a arquitectura aqui representada mais não fosse que a particularização de um dos ramos desse sistema maior.

Na segunda camada encontram-se os elementos funcionais responsáveis pela normalização do formato da informação que chega à plataforma. Todos os elementos desta camada procedem à conversão para o mesmo formato, o que leva à dispensa do elemento de topo de implementar essa funcionalidade. Estes elementos funcionam, por isso, como integradores da informação que pesquisam. Apenas eles conhecem as linguagens de pesquisa e a estrutura e semântica da informação pertencentes às fontes de informação que pesquisam.

Na terceira camada surgem apenas SPRIs com o papel de repositórios de informação ou então sistemas de informação externos (SI Externo), que não fazem parte da plataforma, pertencendo à responsabilidade de entidades exteriores.

A inserção de novos elementos funcionais, que respeitam o modelo subjacente à arquitectura, pode surgir em qualquer das duas últimas camadas, contudo a inserção de elementos exteriores deve sempre ser feita a partir da terceira camada e fazer-se acompanhar pela inserção de um elemento funcional de integração na segunda camada, caso não exista ainda algum que se possa prestar ao papel.

Esta apresentação discriminatória da funcionalidade entre as camadas e as restrições impostas à introdução de novos elementos não devem ser vistas como um desvirtuar do modelo de abstracção, mas sim como uma particularização da implementação de tal modelo. Aliás, esta discriminação não é tão acentuada como parece, uma vez que na

própria arquitectura aparece, por exemplo, um elemento funcional na segunda camada que possui repositório próprio de informação. A própria introdução de elementos exteriores na terceira camada não obriga a que esses elementos possuam repositórios a esse nível.

### 4.4.2 O Elemento Funcional

Conforme o modelo de abstracção, o elemento funcional base nesta arquitectura consiste num sistema que tem por objectivo a pesquisa e a recolha de informação. A arquitectura interna deste sistema encontra-se representada na Figura 4.4.

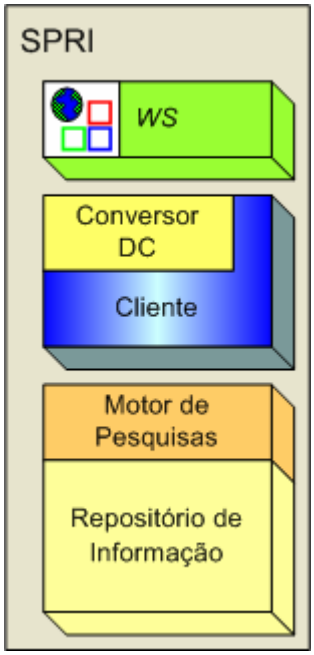



Figura 4.4 – Arquitectura do elemento funcional da plataforma de *middleware*.

A arquitectura do elemento funcional base da plataforma apresenta claramente três partes independentes, correlacionadas com os possíveis papéis que este pode desempenhar:

- a parte inferior, que lhe permite funcionar como um repositório de informação;
- a parte do meio, que lhe permite funcionar como um cliente de repositórios de informação;
- e a parte superior, que lhe permite desempenhar o papel de servidor, apresentando-se como um serviço.

As partes superior e do meio são aquelas que caracterizam o SPRI mais próximo do seu modelo genérico, contudo, e uma vez que a parte inferior implementa também uma forma de pesquisa e recolha de informação, as partes inferior e do meio podem aparecer conjuntamente ou isoladamente, não desvirtuando o modelo subjacente do SPRI.

Concretamente, a parte inferior consiste em dois módulos funcionais que podem tomar a forma de uma base de dados, de qualquer tipo, ou simplesmente de um sistema de ficheiros em associação com uma aplicação que facilite a sua pesquisa. A parte do meio consiste na simbiose entre um módulo funcional que implementa um ou mais clientes para fontes de informação e um módulo, o Conversor DC, que implementa a funcionalidade do subelemento CER, do modelo genérico do SPRI (Figura 4.2), e lhe adiciona a funcionalidade de normalizador de pedidos de pesquisa. A parte superior consiste num módulo que oferece ao exterior a funcionalidade do SPRI na forma de um serviço.

Na arquitectura da plataforma, Figura 4.3, encontram-se representados vários exemplos de SPRIs, que possuem na totalidade ou em parte os módulos aqui descritos.

Na segunda camada, da esquerda para a direita, o primeiro SPRI possui todos os módulos por isso não necessita de pesquisar qualquer outro elemento.

O segundo SPRI possui apenas as partes superior e do meio. Por isso, necessita de enviar pedidos de pesquisa e informação aos dois SPRIs da camada abaixo. Este cenário arquitectural pode ser utilizado em diferentes situações: na situação de um determinado tipo de informação se encontrar distribuído por vários repositórios; ou na situação de a informação se situar simplesmente replicada por vários repositórios, com vista à manutenção de um certo grau de redundância e por consequência ao aumento da fiabilidade do sistema.

Os terceiro e quarto SPRIs possuem também apenas as partes superior e do meio, baseando-se em sistemas de informação externos para aquisição da informação. Estes sistemas externos, apesar de serem eles também sistemas de pesquisa e recolha de informação, não se encontram classificados como SPRIs, no contexto da arquitectura, porque o seus modelos funcional e arquitectural são da responsabilidade de entidades exteriores à plataforma e poderem diferir daqueles que são atribuídos neste trabalho. De resto, estes são bons exemplos da integração de sistemas heterogéneos pela plataforma de *middleware*, em que é possível integrar bibliotecas e sistemas de informação exteriores, usando modelos e tecnologias completamente diversas.

### 4.4.3 A Normalização da Informação

A normalização da informação, ou conversão do formato da informação para um formato comum, é uma questão já ligeiramente abordada nas secções anteriores. A necessidade desta normalização está relacionada com dois dos requisitos colocados à plataforma inicialmente: a necessidade de identificação e remoção de referências de informação duplicadas e a necessidade de integrar, sob o mesmo contexto, modelos de metadados diversos.

O módulo Conversor DC, na parte superior da arquitectura do SPRI, Figura 4.4, é o responsável por esta normalização. De facto, este módulo faz referência, no próprio nome, a uma tecnologia específica utilizada nesta arquitectura: o Dublin Core (DC). O DC consiste num modelo normalizado para a descrição de recursos de informação multi-domínio. Este modelo, na sua perspectiva mais simples, possui apenas quinze elementos de descrição e pode ser usado de duas formas diferentes: primeiro, para catalogar recursos de informação de variadíssimos tipos; segundo, para proceder a pesquisas utilizando os seus termos. Nesta arquitectura está prevista a sua utilização nas suas duas formas.

Quando um pedido de pesquisa é recebido no SPRI, este chega num formato que utiliza os termos do modelo DC. O módulo Conversor DC tem nesse momento a seu cargo a operação de conversão dessa pesquisa para o formato nativo da fonte de informação que o módulo Cliente ou Clientes vai interrogar. Quando, por outro lado, chegam os resultados da pesquisa, estes são de novo encaminhados para o Conversor DC para serem convertidos do seu formato original para o formato DC e averiguada a sua duplicidade face a outros resultados entretanto recebidos. Verifica-se, neste caso, que o módulo Conversor DC possui uma responsabilidade tripla:

- conversão das pesquisas;
- conversão dos resultados;
- identificação e remoção de resultados duplicados.

O modelo Dublin Core é apenas um entre muitos modelos para a descrição de recursos. Qual a razão para o utilizar nesta plataforma e não outro? De facto, este nem sequer é um modelo rico em elementos de descrição, pelo contrário é um modelo bastante simples. O que implica, em muitas conversões, a perda de informação, como por exemplo a conversão do formato MARC, qualquer que seja a sua nuance, para DC.

Realmente, a razão para a adopção deste modelo prende-se com a finalidade que se pretende dar à informação, depois de convertida. A principal finalidade é alcançar um

conjunto de registos descritivos, de formato comum, que facilite a identificação e remoção de informação duplicada, por um lado, e possibilite uma perspectiva globalizante de toda a informação recolhida numa pesquisa, por outro. Para este fim, era necessário um modelo simples que não implicasse posteriores e demoradas operações de identificação e que fosse detentor de elementos descritivos pertinentes que permitissem a rápida e fiel identificação dos itens informativos a que se referem.

Pretende-se através do modelo DC oferecer uma descrição da informação a utilizar, num nível de abstracção superior, sem a imediata disponibilização de pormenores que à primeira vista são dispensáveis. Não se pretende contudo a substituição dos formatos originais por este. O modelo DC pretende apenas ser um meio temporário para a disponibilização de informação sumária. Os registos, no seu formato original, são sempre acessíveis através de requisição posterior.

Ainda em abono da utilização do Dublin Core, pode-se referir a sua adopção pelo protocolo OAI-PMH, que o utiliza de forma obrigatória e numa perspectiva muito próxima da usada neste trabalho. O OAI-PMH é actualmente utilizado por inúmeros repositórios digitais, como atestado na página web da OAI que mantém uma lista actualizada com o registo desses repositórios (OAI, 2009).

### 4.4.4 Uma Arquitectura de Serviços

Os elementos funcionais do modelo de abstracção, descritos antes, receberam o nome de SPRI - Serviço de Pesquisa e Recolha de Informação.

O termo “serviço” é um termo que assume toda a importância no contexto deste trabalho, pois é um termo que se encontra em total consonância com a nomenclatura utilizada no domínio das bibliotecas digitais, para classificação dos agentes participantes numa biblioteca digital (Lagoze and Davis, 1995a; Lagoze et al., 1995b) e, para além disso, este trabalho pode ser classificado, de forma lata, como uma plataforma de integração de serviços de informação.

De facto, não interessa a esta plataforma a origem dos sistemas, o seu ambiente de execução, a sua arquitectura ou mesmo a sua funcionalidade específica. Interessa apenas aquilo que esses sistemas podem oferecer, ou seja, o seu serviço.

Foi na assunção deste paradigma da utilização de serviços que a arquitectura para esta plataforma de *middleware* foi pensada como uma arquitectura de serviços e para tal, foi adoptada a utilização da tecnologia *web services*.

Os *web services* (W3C, 2004b) são actualmente uma das tecnologias mais usadas no âmbito da Internet, para oferecer acesso remoto a sistemas de informação. O seu propósito é aumentar a interoperabilidade entre sistemas heterogéneos, disponibilizando a sua funcionalidade como serviços distribuídos. Este o motivo, que levou à sua adopção como o meio privilegiado para a criação e integração de serviços na presente plataforma de *middleware*.

Conforme é visível na arquitectura do elemento funcional SPRI, Figura 4.4, este elemento possui na sua parte superior um módulo contendo a inscrição WS. Este módulo consiste na implementação do *web service* para o elemento funcional e permite assim a sua integração na plataforma como um serviço.

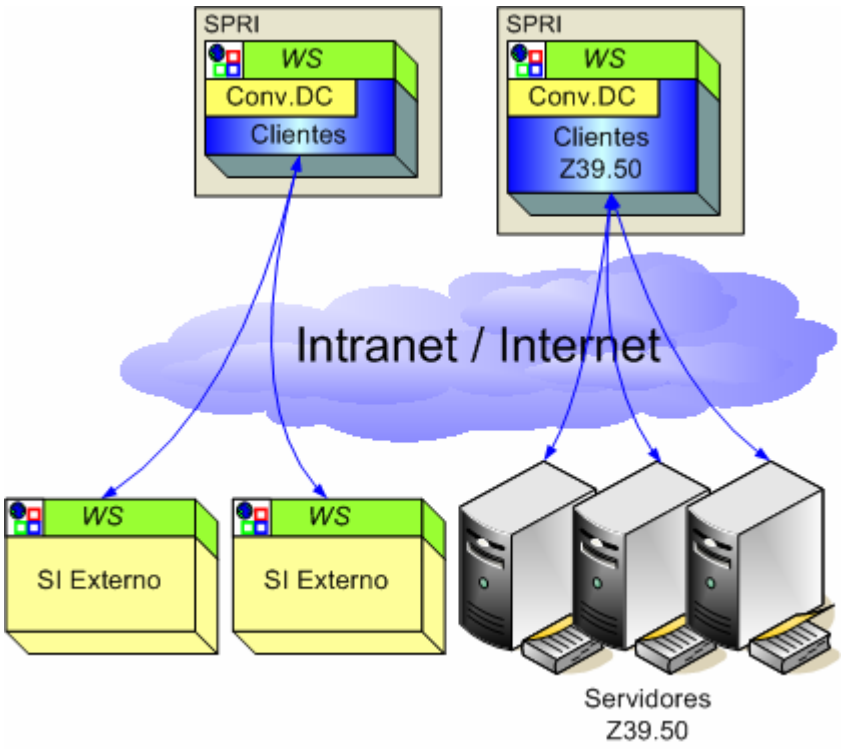



Figura 4.5 – Integração de sistemas exteriores na plataforma.

A integração de sistemas de informação exteriores à plataforma, Figura 4.5, é conseguida através da implementação de SPRIs dedicados que dialogam com esses sistemas na sua linguagem nativa e procedem às devidas conversões para a linguagem da plataforma. Um exemplo paradigmático, e presente na Figura 4.5, é a integração de sistemas de consulta bibliográfica baseados no protocolo Z39.50. Neste caso, é

implementado um SPRI que dialoga com esses sistemas, usando esse protocolo, e procede a todo o trabalho de conversão para a sua integração plena na plataforma.

Outros sistemas exteriores existem que disponibilizam de imediato um ou mais *web services*. A integração desses sistemas é facilitada pela utilização de protocolos comuns, contudo há igualmente um trabalho de conversão a efectuar ao nível da linguagem de pesquisa e da semântica utilizada. Por isso, a integração de tais sistemas deve também ser acompanhada pela implementação de SPRIs dedicados, como ilustra a Figura 4.5.

Os sistemas que implementam apenas repositórios de informação, ver Figura 4.3, e que são tidos como elementos funcionais da plataforma, são integrados na plataforma de uma forma semelhante. Os SPRIs que os pesquisam, podem não necessitar de um trabalho tão apurado de conversão, contudo terão sempre a seu cargo a compreensão da semântica da informação pesquisada.

De forma genérica, é de reter que a integração de qualquer sistema ou serviço na plataforma é efectuada através da sua representação por um SPRI. Seja através de um já existente e que possa dialogar com esse sistema ou serviço, seja através da introdução de um novo.

A tecnologia dos *web services* é utilizada na plataforma de modo transversal. Não se encontra dependente de nenhum sistema nem de nenhum nível funcional particulares. É sim o meio pelo qual é alcançado um elevado nível de interoperabilidade entre diferentes sistemas e contribui também para o aumento da escalabilidade da plataforma.

### 4.4.5 O Directório de Serviços

Um dos requisitos de sistema, colocados à plataforma de *middleware*, é a necessidade de mecanismos que permitam o "registo das características e da localização dos sistemas a aceder". Ou seja, existe a necessidade de manter um registo de todos os serviços disponíveis para utilização na plataforma, no qual é guardada informação acerca do tipo de serviço, o seu provedor, a sua localização na rede, etc.

Os meios para a implementação de tal registo, que é considerado neste contexto um directório de serviços, são vários, desde o uso de bases de dados até à utilização de simples ficheiros de configuração. Contudo, depois da adopção dos *web services* foi decidido manter a fidelidade a essa tecnologia e adoptar a tecnologia UDDI, expressamente concebida para o registo, descrição e localização de *web services*.

Na Figura 4.6 encontra-se representada uma parte da arquitectura da plataforma de *middleware* em que sobressai a interacção entre os SPRIs e o directório de serviços.

Apesar de na Figura 4.3 se encontrarem representados dois directórios, o directório é único. Tal representação deve-se unicamente a uma maior legibilidade da figura.

O directório de serviços da plataforma de *middleware* consiste num directório proprietário à plataforma, embora implementado utilizando tecnologias abertas. Não há inicialmente intenção de abrir esse directório ao domínio público, pois este tem como missão apenas manter um registo de todos os serviços a serem utilizados na plataforma, nas diferentes camadas de abstracção. Contudo, se essa intenção se vier a verificar, não haverá problemas de ordem técnica, uma vez que a interface para o directório é bem conhecida no domínio público.

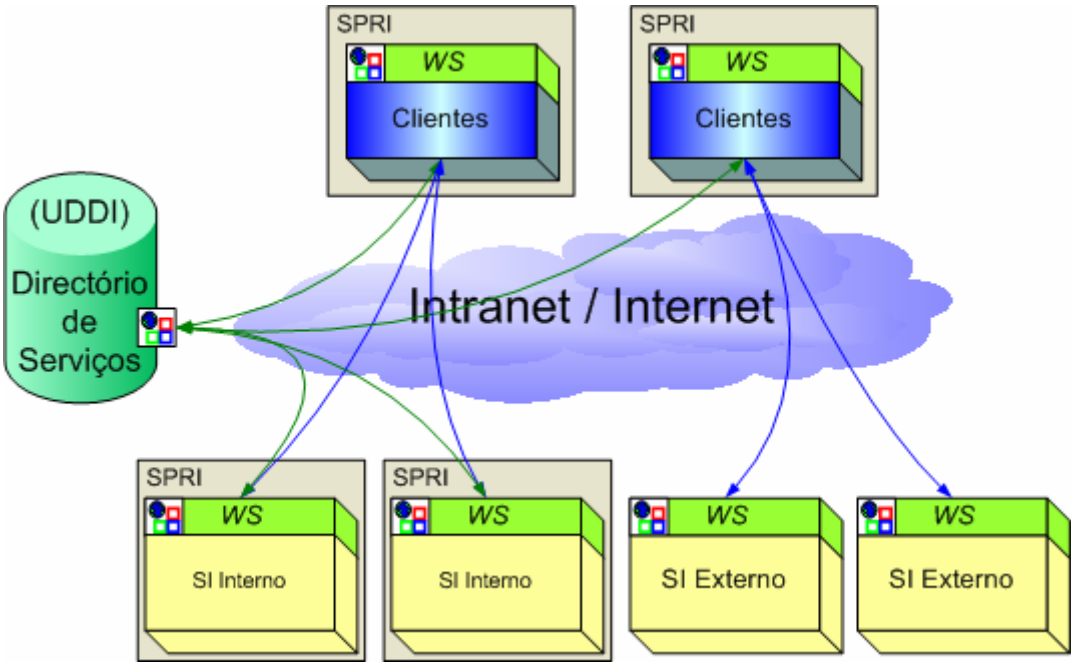



Figura 4.6 – O directório de serviços.

O acesso ao directório de serviços é efectuado com dois propósitos distintos: no primeiro, para publicar os dados relativos aos serviços; no segundo, para consultar esses mesmos dados.

Quando um serviço (SPRI) inicia o seu funcionamento na plataforma, este começa por registar os seus dados no directório. Depois disso, apenas voltará a aceder o directório com o objectivo de escrita se os seus dados se alterarem, como a mudança da sua localização por exemplo, ou se o serviço for desactivado, o que levará à eliminação dos dados do serviço. Este cenário não se verifica para serviços de informação externos (SI

Externos), como é óbvio, pois estes não conhecem o directório da plataforma. Neste caso concreto, o registo desses serviços no directório terá de ser efectuado por outros meios: manualmente, através da intervenção directa do administrador do sistema, ou por meios automáticos, utilizando uma aplicação que proceda à consulta de directórios de serviços externos e faça a actualização.

Os acessos de consulta servem para a descoberta de serviços que se enquadram dentro da política de pesquisa de cada um dos SPRIs e essencialmente para confirmação da localização desses mesmos serviços. Em funcionamento normal, um SPRI que pretende aceder a outro deve, antes de mais, pesquisar o directório para confirmar a existência e localização do serviço e só depois proceder ao acesso. Claro que a obrigação de preceder cada acesso a um serviço por um acesso de confirmação ao directório gera obviamente latência no sistema. Para isso deve ser implementado um mecanismo de *caching* em cada SPRI, por forma a que essa consulta seja efectuada apenas periodicamente ou quando forem detectadas anomalias no acesso ao serviço, como a sua não localização, ou a inexistência de funcionalidades anteriormente existentes, por exemplo.

## 4.5 A Interface Funcional Comum

Os elementos que instanciam, na arquitectura, os SPRIs das duas primeiras camadas do modelo de abstracção da plataforma, são os elementos que constituem o cerne da plataforma de *middleware*, como já explícito anteriormente. Os elementos da segunda camada, podem ser detentores de funcionalidades bastante diversas, tendo em vista que têm por missão a integração de sistemas e dados diversos. Contudo, como elementos agregadores que são, com a subsequente necessidade de comunicarem os seus resultados ao elemento único da primeira camada, devem possuir uma parte da funcionalidade que obedeça a um mínimo denominador comum. Esta imposição, de uma funcionalidade comum, embora não absolutamente indispensável, tem a vantagem de permitir o acesso de todos os elementos utilizando a mesma interface e contribui para uma fácil e eficaz escalabilidade da plataforma.

Desta forma, foi concebida uma interface funcional minimalista com a qual todos os SPRIs mencionados acima são conformes, inclusivamente o SPRI da primeira camada. A sua representação encontra-se na Figura 4.7 e apresenta o nome SpriInterface.

A interface SpriInterface apresenta um máximo de dez métodos (operações) e o seu espectro funcional vai desde a simples verificação de existência do serviço até à recolha

de objectos multimédia e índices. A lista real das operações oferecidas nos SPRIs pode ser muito mais vasta e diversificada, contudo esta é a lista mínima que se comprometem a implementar e por isso deve sempre encontrar eco por parte de um elemento cliente, mesmo quando a operação não efectua qualquer trabalho útil no contexto de um determinado SPRI.

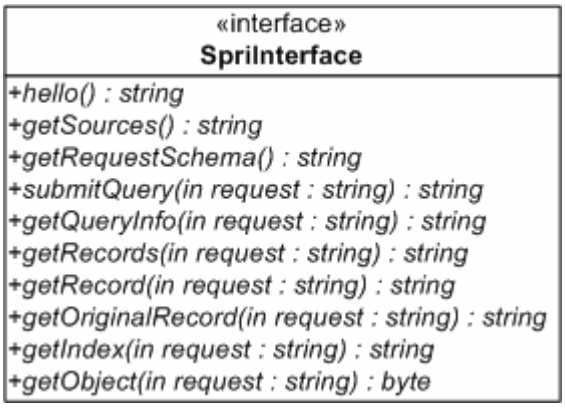



Figura 4.7 – A interface funcional comum.

De seguida procede-se a uma explanação da funcionalidade de cada um dos métodos exibidos nesta interface.

## 4.5.1 Os Métodos da Interface

### 4.5.1.1 O método hello()

O método hello() tem por objectivo principal a verificação da disponibilidade do serviço. Quando invocada, responde sempre com a palavra “hello”. Desta forma, a resposta deste método indica a disponibilidade do serviço e a ausência de resposta indica a indisponibilidade do mesmo.

A indisponibilidade do serviço pode ser devida a diferentes situações: o não funcionamento do serviço; a mudança do serviço para outra localização; a sobrecarga do serviço e por isso a consequente falta de resposta.

Quando a indisponibilidade do serviço é verificada por um cliente, este deve consultar de novo o directório de serviços para confirmar a sua existência e localização. A indisponibilidade devida ao não funcionamento do serviço ou a eventuais problemas a ocorrer com o serviço não são passíveis de resolução por parte do cliente.

#### 4.5.1.2 O método getSources()

O método getSources() tem por objectivo devolver informação acerca das fontes de informação passíveis de serem consultadas. A pertinência desta informação para os clientes do serviço relaciona-se com a precisão que é possível utilizar nas consultas a efectuar ao sistema.

#### 4.5.1.3 O método getRequestSchema()

Os três métodos descritos até ao momento, não aceitam qualquer parâmetro, conquanto os seus nomes identificam perfeitamente a funcionalidade que se espera deles. Contudo, todos métodos descritos a seguir, aceitam um, e apenas um só, parâmetro de nome *request* e do tipo *string*. Não nos deixemos enganar pela simplicidade deste procedimento. Este parâmetro único consiste na realidade num documento XML contendo toda a parametrização do pedido, podendo tornar bastante complexo o processo de parametrização, com a possibilidade de o número real de parâmetros ser variável.

Tendo isto em mente, e visto que o documento XML de parametrização tem de obrigatoriamente obedecer a uma estrutura que será entendida pelos métodos, foi concebido um XML *Schema* que permite a sua validação aquando da sua recepção pelos mesmos. O método getRequestSchema() devolve esse *schema* que permite proceder à criação , por parte dos clientes, de documentos *request* com a estrutura esperada.

#### 4.5.1.4 O método submitQuery()

O método submitQuery() tem por objectivo a submissão de um pedido de pesquisa ao SPRI em questão. Esse pedido será distribuído pelos SPRIs da camada abaixo, conforme eles existam e conforme a parametrização do pedido em relação às fontes a procurar. O resultado devolvido por este método será um documento XML, onde constará um registo específico de informação de estado sobre a pesquisa e um conjunto de registos DC que satisfazem o pedido.

#### 4.5.1.5 O método getQueryInfo()

O método getQueryInfo() permite pedir especificamente, e apenas, o registo de informação de estado relativo a um pedido de pesquisa previamente efectuado. Este método é tanto mais importante devido à dinâmica do processamento das pesquisas. Após a recepção dos primeiros resultados, em resposta a um pedido de pesquisa pelo método submitQuery(), o processamento da pesquisa pode não se encontrar terminado o

que levará com certeza a que a informação de estado dessa pesquisa continue a mudar até ao seu término. Desta forma, é particularmente importante que o cliente que pediu a pesquisa tenha a possibilidade de saber mais relativamente ao progresso da mesma.

#### 4.5.1.6 O método getRecords()

O método getRecords() é um método complementar ao método submitQuery(). Não é desejável, nem aceitável, que este último método espere por todo o desenrolar de uma pesquisa e apenas no fim devolva todos os resultados encontrados. Este procedimento num cenário de pesquisa distribuída pode retumbar num completo fracasso, tanto do ponto de vista dos sistemas, que poderão não suportar tal carga, como do ponto de vista do utilizador, que poderia ter de esperar horas para obter algum resultado. Em vez disso, o método submitQuery() devolve em tempo útil (parametrizável) alguns resultados, ficando o método getRecords() responsável por recolher posteriores resultados, à medida das necessidades do cliente.

#### 4.5.1.7 O método getRecord()

O método getRecord() permite a recolha de um registo específico. Este método é particularmente útil em situações em que o cliente não faz *caching* dos resultados obtidos e desta forma pretende aceder a um determinado registo com vista à sua apresentação.

#### 4.5.1.8 O método getOriginalRecord()

O método getOriginalRecord() possibilita a recolha do registo original relativo ao registo em DC que o representa. De facto, o documento devolvido por este método consiste na combinação entre XML e XSLT, por forma a devolver o registo e a sua representação, sempre que possível. A razão para tal, prende-se com o facto de a plataforma de *middleware* não conhecer nem compreender, ao seu detalhe, os registos originais recolhidos. Contudo, se o cliente da plataforma detiver esse conhecimento poderá sempre optar por outra forma de representação do registo.

#### 4.5.1.9 O método getIndex()

O método getIndex() permite a recolha de listas de valores únicos existentes para cada índice pesquisável. Pode ser por exemplo: uma lista de autores, uma lista de assuntos; uma lista de títulos, etc. O interesse desta informação por parte de um cliente, reside na possibilidade deste percorrer a informação ao invés de apenas poder pesquisar por termos, que desconhece inicialmente se existem ou não.

#### 4.5.1.10 O método getObject()

O método getObject() é o último nesta interface, mas não o menos importante. Este método permite a recolha de objectos binários (informação multimédia – textos, imagens, vídeos, sons, programas, etc.) que se encontrem referenciados nos registos recolhidos.

### 4.5.2 A modelação do *Web Service*

Na secção anterior foram descritos os papéis dos métodos pertencentes à interface funcional comum, depois de na Figura 4.7 se ter representado, de forma sumária, essa mesma interface num diagrama de interface baseado em UML (OMG, 2009). Não foram apresentadas ligações a outros elementos porque a interface é igual para todos os elementos intervenientes e as suas relações foram já devidamente clarificadas no modelo de abstracção genérico para a plataforma de *middleware* na Figura 4.1.

Contudo, os elementos que compõem a plataforma, os SPRIs, foram já anteriormente apresentados como serviços, mais propriamente como *web services*. A linguagem WSDL, propositadamente concebida para descrever este tipo de serviços, acaba por também desempenhar um papel importante na sua concepção e modelação. Daí que seja apresentada, nesta secção, a modelação do serviço, utilizando este padrão.

Recorrendo a uma aplicação comercial, o Altova XMLSpy (Altova, 2009), foi gerada a representação esquemática do modelo WSDL da interface. Na Figura 4.8 encontra-se representado esse esquema.

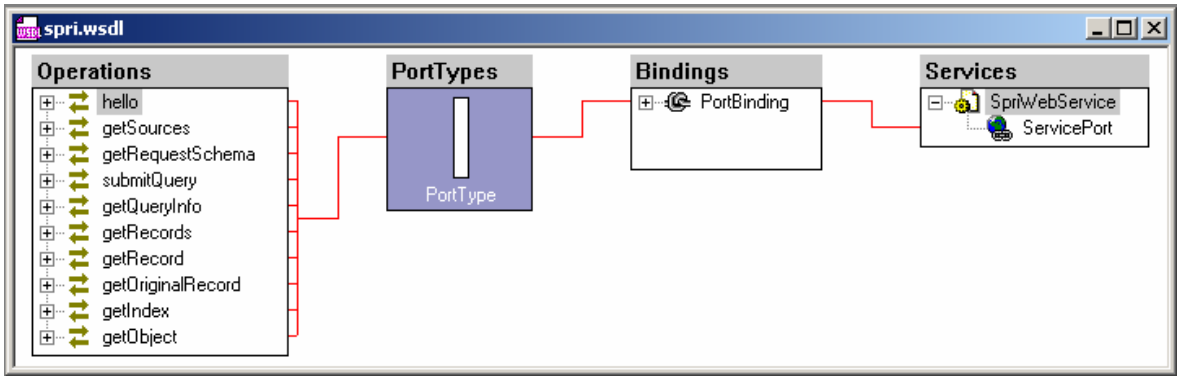



Figura 4.8 – O modelo WSDL da interface funcional comum de um SPRI.

Da direita para a esquerda, encontra-se no primeiro elemento diagramático a designação do serviço, “SpriWebService”, e o porto, “ServicePort”, utilizado por este. Na Figura 4.9 encontra-se a sua definição em WSDL.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions ... >
  ...
  <service name="SpriWebService">
    <documentation>SPRI Main Web Service</documentation>
    <port name="ServicePort" binding="impl:PortBinding">
      <soap:address location="http://www.ua.pt/spri.asmx"/>
    </port>
  </service>
</definitions>
```

Figura 4.9 – Definição do serviço “SpriWebService”.

Na definição do serviço em WSDL é visível a localização do serviço através do seu porto.

Tanto no diagrama, na Figura 4.8, como na definição, Figura 4.9, é visível a ligação do porto “ServicePort” a outro elemento, o “PortBinding”. A definição deste elemento encontra-se parcialmente reproduzida na Figura 4.10 e define a forma de transporte a utilizar pelas mensagens do serviço. Neste caso é utilizado o protocolo SOAP orientado ao documento, para transporte das mensagens específicas do serviço, e utiliza o protocolo HTTP para transporte das próprias mensagens SOAP.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions ... >
  ...
  <binding name="PortBinding" type="impl:PortType">
    <soap:binding style="document"
          transport="http://schemas.xmlsoap.org/soap/http"/>
    ...
  </binding>
  ...
</definitions>
```

Figura 4.10 – Definição do elemento PortBinding.

Este elemento encontra-se, por sua vez, também ligado a outro elemento diagramático, o “PortType”, no qual se encontram definidas as operações suportadas pelo serviço. A sua definição encontra-se parcialmente reproduzida na Figura 4.11.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions ... >
 ...
 <portType name="PortType">
  <operation name="hello">
   <input message="impl:Null"/>
   <output message="impl:Response"/>
  </operation>
  ...
  <operation name="submitQuery">
   <input message="impl:Request"/>
   <output message="impl:Response"/>
  </operation>
  ...
  <operation name="getObject">
   <input message="impl:Request"/>
   <output message="impl:BinaryResponse"/>
  </operation>
 </portType>
 ...
</definitions>
```

Figura 4.11 – Definição do “PortType” do serviço.

O elemento diagramático “Operations”, Figura 4.12, apesar de aparecer no diagrama como um elemento independente, encontra a sua definição inclusa no elemento “PortType”. Este elemento representa o conjunto completo de operações disponíveis no serviço, assim como as mensagens de entrada (*input*) e saída (*output*) que as implementam. Todas as mensagens de entrada que não possuem parâmetro, são do tipo “impl:Null”, e todas as que possuem, são do tipo “impl:Request”. As mensagens de saída são todas do tipo “impl:Response”, com excepção para a mensagem de saída da operação “getObject” que pertence ao tipo “impl:BinaryResponse”.

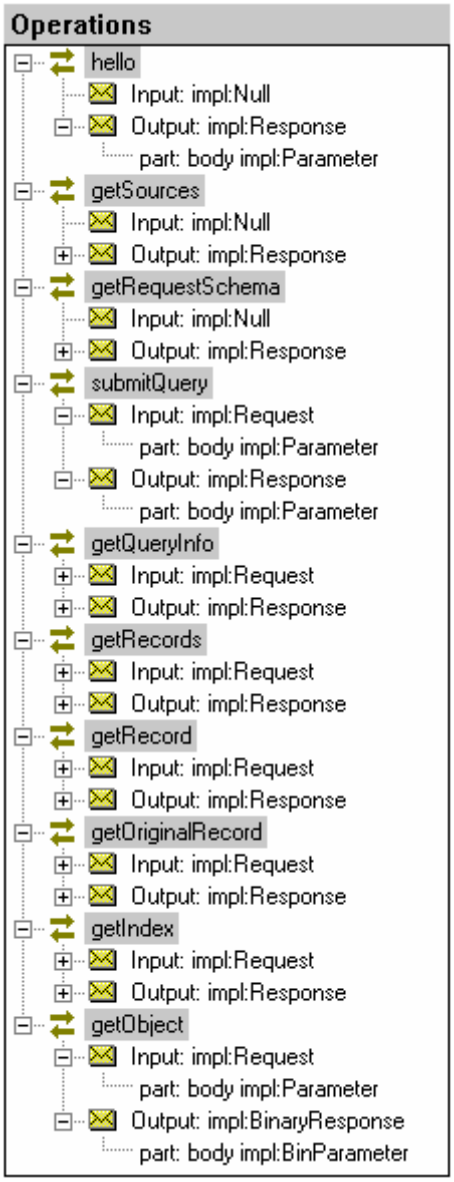



Figura 4.12 – Diagrama das operações do serviço.

A definição destas mensagens, assim como dos parâmetros que contêm, são definidas através dos elementos “message” e dentro do elemento “types”, que permite definir um *schema* para definição completa dos parâmetros. Na Figura 4.13 encontra-se ilustrada esta definição.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions ... >
 <types>
  <xs:schema targetNamespace="urn:spri:webservice">
   <xs:element name="Parameter" type="xs:string"/>
   <xs:element name="BinParameter" type="xs:base64Binary"/>
  </xs:schema>
 </types>
 <message name="Null"/>
 <message name="Request">
  <part name="body" element="impl:Parameter"/>
 </message>
 <message name="Response">
  <part name="body" element="impl:Parameter"/>
 </message>
 <message name="BinaryResponse">
  <part name="body" element="impl:BinParameter"/>
 </message>
 ...
</definitions>
```

Figura 4.13 – Definição das mensagens e dos seus parâmetros.

Verifica-se então, através da análise da definição ilustrada na Figura 4.13, que são definidas apenas quatro mensagens base (Null, Request, Response e BinaryResponse) para implementação das dez operações e são definidos apenas dois tipos distintos de parâmetros para essas mensagens (Parameter e BinParameter).

Esta definição consiste numa definição optimizada, apenas possível através de uma cuidada análise das necessidades para cada operação e da sua definição manual. O processo automatizado para esta definição, passível de ser utilizado em ferramentas como o XMLSPY, gera uma quantidade muito maior de mensagens e parâmetros diferentes, que sobrecarrega a modelação de um *web service*, ao ponto de por vezes se tornar difícil a compreensão do seu propósito.

## 4.6 Os Modelos de Dados

Na secção anterior foi descrita a interface mínima comum que deve pautar o comportamento funcional dos SPRIs nativos à plataforma de *middleware*. Apesar da descrição pormenorizada dessa interface, muito pouco foi dito sobre a natureza dos dados que entram e saem desses serviços.

Na presente secção pretende-se dar a conhecer, de forma também pormenorizada, os modelos que regem os pedidos passíveis de efectuar a esses serviços, assim como os modelos que regem as suas respostas.

### 4.6.1 O Modelo de Dados dos Pedidos

Como já abordado na secção 4.5.1 - Os Métodos da Interface, os métodos que aceitam parâmetros aceitam apenas um. Este parâmetro consiste num documento XML que contem toda a parametrização do pedido. Para isso é necessário que esse documento obedeça a regras, no sentido de ser entendido pelo serviço que detém o método. Com esse intuito, foi concebido um modelo, na forma de um XML *Schema*, que permite, tanto a validação do documento por parte da entidade receptora como a própria criação do documento por parte da entidade emissora.

#### 4.6.1.1 O Elemento request

O elemento "request" é o elemento raiz de qualquer documento de pedido. É aquele que define à partida o documento como um pedido de informação. Podendo esse pedido ser variado, conforme o método a chamar no serviço.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
targetNamespace="http://www.ua.pt/spri/requests"
xmlns:ua="http://www.ua.pt/spri/requests"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
...
  <xsd:element name="request">
    <xsd:complexType>
      <xsd:choice>
        <xsd:element ref="ua:query"/>
        <xsd:element ref="ua:info"/>
        <xsd:element ref="ua:records"/>
        <xsd:element ref="ua:record"/>
        <xsd:element ref="ua:index"/>
        <xsd:element ref="ua:object"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

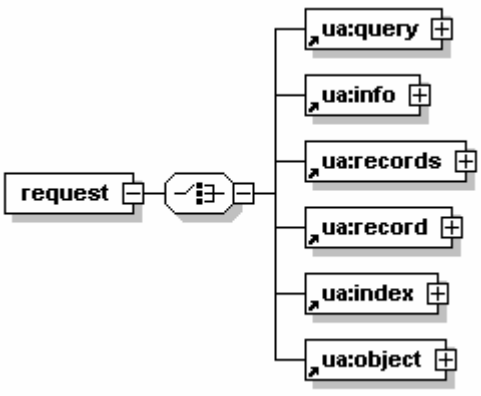



Figura 4.14 – XML *Schema* e diagrama do elemento "request".

Na Figura 4.14 encontram-se ilustrados o XML *Schema* e o correspondente diagrama, que permitem a visualização da possível composição do elemento “request”. A notação do diagrama é a utilizada pela aplicação XMLSPY para a representação gráfica de um XML *Schema*.

Este elemento, assim como todos aqueles que são inclusos nele, pertencem ao *namespace* “ua”, o que leva a que na representação em diagrama os elementos apareçam umas vezes com o prefixo do seu *namespace*, outras vezes sem, dependendo do contexto em que aparecem.

O elemento “request” poderá conter um, e apenas um só, dos seis elementos visíveis na figura, com o objectivo de criar um documento que seja aceite por um dos métodos do serviço. Por isso, a inclusão dos seis elementos no *schema* é precedido de um elemento “choice” que obriga à escolha de um dos elementos para instanciar o documento XML de pedido a partir do modelo.

#### 4.6.1.2 O Elemento query

O elemento “query” permite criar um documento para submeter um pedido de pesquisa ao serviço, através do método submitQuery(). O seu diagrama encontra-se representado na Figura 4.15.

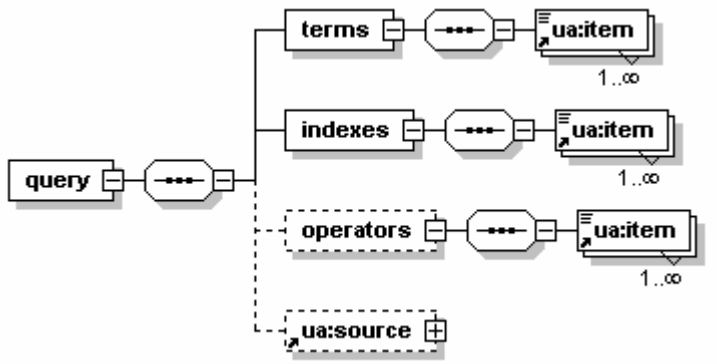



Figura 4.15 – Diagrama do elemento “query”.

Este elemento é composto por outros quatro elementos: dois obrigatórios (“terms” e “indexes”) e dois opcionais (“operators” e “source”). O elemento “terms” permite abrigar uma lista de termos diversos a pesquisar. O elemento indexes permite abrigar uma lista de índices a serem utilizados na pesquisa dos termos. Para cada termo especificado na lista de termos, deve existir um índice na lista de índices. O método utilizado para associar cada termo e o seu índice de pesquisa é o método sequencial: ao primeiro termo

está associado o primeiro índice, ao segundo termo está associado o segundo índice, e assim de seguida.

O elemento “operators” permite abrigar uma lista de operadores lógicos que podem possuir os valores: e, ou, e não; e são utilizados para definir a relação entre os múltiplos termos de pesquisa. Este elemento aparece como opcional no modelo apenas para contemplar a situação de a operação de pesquisa se basear num único termo. Situação em que não faz sentido a especificação de operadores. Quando o número de termos é superior a um, então o pedido de pesquisa deve obrigatoriamente contemplar uma lista de operadores, que em número deve sempre ser inferior em uma unidade ao número de termos.

O elemento “source”, também opcional, permite especificar a fonte ou as fontes de pesquisa às quais é dirigida a pesquisa. Este elemento possui dois atributos, que não aparecem representados no diagrama: um obrigatório, “uri”, e outro opcional, “context”. O atributo “uri” permite a identificação da fonte; o atributo “context” permite especificar se a pesquisa é dirigida a todas as eventuais fontes dependentes desta ou apenas às fontes especificadas no elemento. Os possíveis valores para este atributo são: “all” – todas as fontes; “restricted” – apenas as fontes especificadas. Para além destes atributos, este elemento é composto por vários outros elementos, como representado na Figura 4.16.

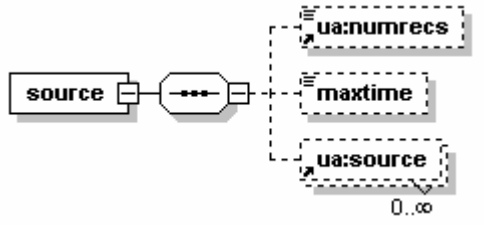



Figura 4.16 – Diagrama do elemento “source”.

Todos os elementos que compõem o elemento “source” são opcionais. Poderá conter o elemento “numrecs” para indicar o número de registos a recolher. Se tal não se verificar, será utilizado um número configurado por defeito no sistema. Poderá também ser indicado o tempo máximo, em segundos, para a realização da pesquisa, através do elemento “maxtime”. Após esse tempo, serão devolvidos os resultados obtidos até ao momento. À imagem do elemento anterior, se este não for especificado, será utilizado um valor por defeito. Opcional é também a especificação de fontes de pesquisa, que se encontram na dependência da primeira. Através da introdução múltipla e recursiva do

próprio elemento “source”, é possível parametrizar, até ao limite, as fontes e subfontes a pesquisar, utilizando os elementos numrecs e maxtime para cada uma delas.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:request xmlns:ua="http://www.ua.pt/spri/requests">
 <ua:query>
  <ua:terms>
   <ua:item>java</ua:item>
   <ua:item>john</ua:item>
   <ua:item>programming</ua:item>
  </ua:terms>
  <ua:indexes>
   <ua:item>dc:title</ua:item>
   <ua:item>dc:creator</ua:item>
   <ua:item>dc:subject</ua:item>
  </ua:indexes>
  <ua:operators>
   <ua:item>and</ua:item>
   <ua:item>or</ua:item>
  </ua:operators>
  <ua:source uri="urn:spri:main" context="restricted">
   <ua:numrecs>100</ua:numrecs>
   <ua:maxtime>60</ua:maxtime>
   <ua:source uri="urn:ar:archive">
    <ua:numrecs>50</ua:numrecs>
   </ua:source>
   <ua:source uri="urn:ua:archive">
    <ua:numrecs>50</ua:numrecs>
   </ua:source>
   <ua:source uri="urn:ua:opacs" context="all">
    <ua:numrecs>50</ua:numrecs>
    <ua:source uri="urn:opacs:brunel">
     <ua:numrecs>20</ua:numrecs>
    </ua:source>
   </ua:source>
  </ua:source>
 </ua:query>
</ua:request>
```

Figura 4.17 – Exemplo de um documento de pedido de pesquisa.

Na Figura 4.17 é apresentado um exemplo de um documento de pedido de pesquisa. Neste exemplo a pesquisa consiste em procurar elementos que possuam no seu título (dc:title) o termo “java” e que possuam também como autor (dc:author) um nome que contenha o termo “john” ou então que esteja classificado no assunto (dc:subject) “programming”. Este pedido de pesquisa é destinado à fonte de informação principal (urn:spri:main), contemplando apenas algumas das suas subfontes. Neste caso, todos os SPRIs são vistos como fontes ou subfontes de informação. A fonte de informação

principal é o primeiro SPRI da plataforma que depois distribui a pesquisa pelos SPRIs da segunda camada de abstracção, tendo em conta a especificação das subfontes no pedido de pesquisa.

O conceito de recursividade introduzido na especificação das fontes de informação permite a integração de fontes, ou SPRIs, com identificadores iguais, o que nunca deveria acontecer. Contudo, se tal se verificar, tal não consiste um problema pois a plataforma baseia-se nos identificadores das fontes em conjunto com a sua hierarquia na integração, para a sua identificação. Apenas há uma restrição, não poderão aparecer dois SPRIs com o mesmo identificador ligados ao mesmo SPRI da camada de abstracção superior. Nesse caso seria impossível a resolução da ambiguidade.

#### 4.6.1.3 O Elemento info

O elemento “info” permite criar um documento de pedido de informação de estado sobre uma pesquisa previamente executada. Este documento destina-se a valor do parâmetro do método getQueryInfo(). Na Figura 4.18 encontra-se representado o diagrama da composição deste elemento e na Figura 4.19 encontra-se um exemplo de um documento contendo este elemento.

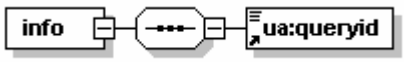



Figura 4.18 – Diagrama do elemento “info”.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:request xmlns:ua="http://www.ua.pt/spri/requests">
  <ua:info>
    <ua:queryid>query123456789</ua:queryid>
  </ua:info>
</ua:request>
```

Figura 4.19 – Exemplo de um documento de pedido de informação de estado.

O elemento “info” contém apenas um elemento, o elemento “queryid”, que permite identificar a pesquisa sobre a qual se pretende obter a informação de estado.

#### 4.6.1.4 O Elemento records

O elemento “records” permite criar um documento de pedido de registos, relativamente a uma pesquisa já efectuada. Este documento destina-se a valor do parâmetro do

método getRecords(). Na Figura 4.20 encontra-se representado o diagrama da composição deste elemento.

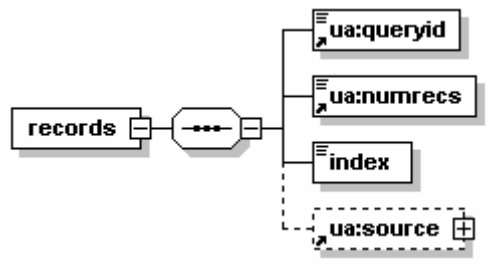



Figura 4.20 – Diagrama do elemento “records”.

O elemento “records” é composto por: um elemento “queryid”, para identificação da pesquisa em relação à qual se pretende recolher registos: um elemento numrecs, para indicar a quantidade de registos pretendidos; um elemento índex, para identificar o número de registo a partir do qual devem ser recolhidos os registos; e opcionalmente, um elemento “source” para eventualmente identificar a origem dos registos. Caso este último elemento não se encontre presente, os registos são recolhidos indiferenciadamente; caso contrário, será primeiro criado um subconjunto de registos pertencentes apenas às origens especificadas e só então serão seleccionados os registos pretendidos.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:request xmlns:ua="http://www.ua.pt/spri/requests">
 <ua:records>
  <ua:queryid>query123456789</ua:queryid>
  <ua:numrecs>10</ua:numrecs>
  <ua:index>21</ua:index>
  <ua:source uri="urn:spri:main">
   <ua:source uri="urn:ua:archive"/>
   <ua:source uri="urn:ua:opacs">
    <ua:source uri="urn:opacs:brunel"/>
   </ua:source>
  </ua:source>
 </ua:records>
</ua:request>
```

Figura 4.21 – Exemplo de um documento de pedido de registos.

Na Figura 4.21 encontra-se um exemplo de um documento contendo este elemento. Neste caso específico são pedidos dez registos, a partir do vigésimo primeiro registo, inclusive, da pesquisa identificada por “query123456789”. Os registos devem pertencer

às fontes “urn:ua:archive” ou “urn:ua:opacs” dentro da “urn:spri:main” e à fonte “urn:opacs:brunel” dentro da “urn:ua:opacs”.

#### 4.6.1.5 O Elemento record

O elemento “record” permite criar um documento para pedido de um registo específico, relativamente a uma pesquisa já efectuada. Este documento destina-se ao valor do parâmetro dos métodos getRecord() e getOriginalRecord(). Na Figura 4.22 encontra-se representado o diagrama da composição deste elemento.

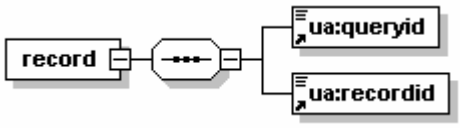



Figura 4.22 – Diagrama do elemento “record”.

Este elemento é composto por um elemento queryid, para identificação da pesquisa em relação à qual se pretende recolher um registo, e por um elemento “recordid”, que identifica univocamente o registo no conjunto de registos resultante da pesquisa.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:request xmlns:ua="http://www.ua.pt/spri/requests">
  <ua:record>
    <ua:queryid>query123456789</ua:queryid>
    <ua:recordid>21</ua:recordid>
  </ua:record>
</ua:request>
```

Figura 4.23 – Exemplo de um documento de pedido de registo.

Na Figura 4.23 encontra-se representado um exemplo de pedido de registo. Quando é utilizado o método getRecord(), é devolvido um registo sumário em DC, quando é utilizado o método getOriginalRecord(), é devolvido um documento que contém o registo original e a sua representação, na forma de XSLT.

#### 4.6.1.6 O Elemento index

O elemento “index” permite criar um documento para pedido de listas de valores de um ou vários índices. Este documento destina-se a valor do parâmetro do método

getIndex(). Na Figura 4.24 encontra-se representado o diagrama da composição deste elemento.

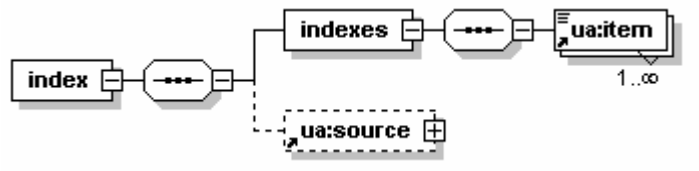



Figura 4.24 – Diagrama do elemento “index”.

Este elemento é composto por um elemento “indexes” obrigatório e por um elemento “source” opcional. O elemento indexes permite a introdução de uma lista de índices, para recolher um conjunto de valores para cada um. O elemento “source”, à imagem do que sucede noutros elementos descritos anteriormente, permite especificar as fontes de informação de onde devem ser recolhidos os valores.

Na Figura 4.25 encontra-se um exemplo de pedido de vários índices (autor, data e assunto) que deverão ser recolhidos apenas de duas fontes de informação “urn:ua:archive” e “urn:ar:archive”.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:request xmlns:ua="http://www.ua.pt/spri/requests">
 <ua:index>
  <ua:indexes>
   <ua:item>dc:author</ua:item>
   <ua:item>dc:date</ua:item>
   <ua:item>dc:subject</ua:item>
  </ua:indexes>
  <ua:source uri="urn:spri:main">
   <ua:source uri="urn:ua:archive"/>
   <ua:source uri="urn:ar:archive"/>
  </ua:source>
 </ua:index>
</ua:request>
```

Figura 4.25 – Exemplo de um documento de pedido de índices.

#### 4.6.1.7 O Elemento object

O elemento “object” permite criar um documento para pedido de um objecto multimédia, referenciado num registo. Este documento destina-se a valor do parâmetro

do método getObject(). Na Figura 4.26 encontra-se representado o diagrama da composição deste elemento.

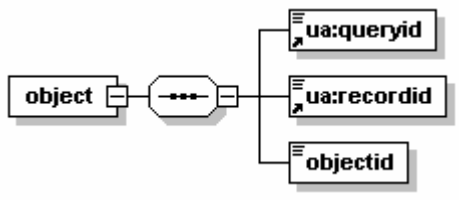



Figura 4.26 – Diagrama do elemento “object”.

Este elemento é composto por um elemento queryid e um elemento recordid, para identificação do registo de referência, e por um elemento “objectid”, que identifica o objecto no contexto do registo.

Na Figura 4.27 encontra-se representado um exemplo de pedido de objecto.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:request xmlns:ua="http://www.ua.pt/spri/requests">
  <ua:object>
    <ua:queryid>query123456789</ua:queryid>
    <ua:recordid>21</ua:recordid>
    <ua:objectid>image003</ua:objectid>
  </ua:object>
</ua:request>
```

Figura 4.27 – Exemplo de um documento de pedido de objecto.

## 4.6.2 O modelo de Dados das Respostas

As respostas devolvidas pelos métodos da interface do serviço são também elas documentos XML contendo os resultados encontrados. Estes documentos, são, também eles, instanciados a partir de um modelo na forma de um XML *Schema*, que é descrito de seguida. Os métodos utilizados para a sua descrição são os mesmos que foram utilizados na descrição do modelo dos pedidos.

### 4.6.2.1 O elemento results

O elemento “results” é o elemento raiz de qualquer documento de resposta. A sua composição pode contudo ser diversa, em função da informação a transmitir.

Na Figura 4.28 encontra-se ilustrado o diagrama de composição do elemento “results”.

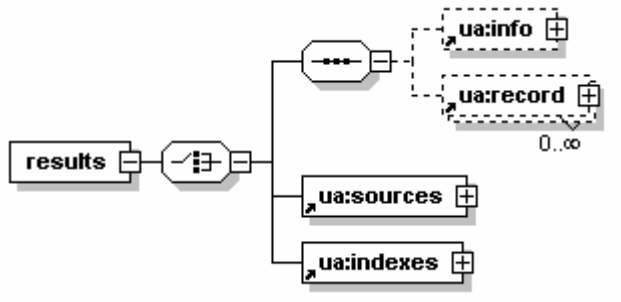



Figura 4.28 – Diagrama do elemento “results”.

O elemento “results” poderá dar origem a três tipos elementares de documentos de resposta, consoante a composição deste. Num tipo, o elemento poderá ser composto pelos elementos “info” e “record”; noutro, composto por um elemento “sources”; e noutro, composto pelo elemento “indexes”. Os elementos que dão origem a estes três tipos de documentos não poderão nunca aparecer em simultâneo.

#### 4.6.2.2 Os elementos info e record

Os elemento “info” aparece sempre nos documentos de resposta dos métodos: submitQuery(), getQueryInfo(), getRecords(). O seu diagrama de composição encontra-se representado na Figura 4.29. Este elemento é composto por: um elemento “queryid”, que identifica a pesquisa; um elemento “query”, que transporta uma versão sumária e textual da pesquisa efectuada; um elemento “status”, que informa o estado de processamento da pesquisa, podendo conter apenas os valores “terminated” e “running”; e um elemento “records”, que é composto por outros elementos e fornece informação acerca dos registos pesquisados.

O elemento “records” é composto por um elemento “hits”, que por sua vez é composto por um elemento “total” e poderá conter um elemento “source”. Este elemento “hits”, permite fornecer informação acerca do número de registos encontrados no sistema, para a pesquisa em questão. O elemento “total”, informa do número total de registos e o elemento “source” permite fragmentar essa informação por fonte.

O elemento “total”, opcional, pertencente ao elemento “records”, fornece informação acerca do número total de registos recolhidos. O qual é em geral bastante inferior ao valor de” hits”.

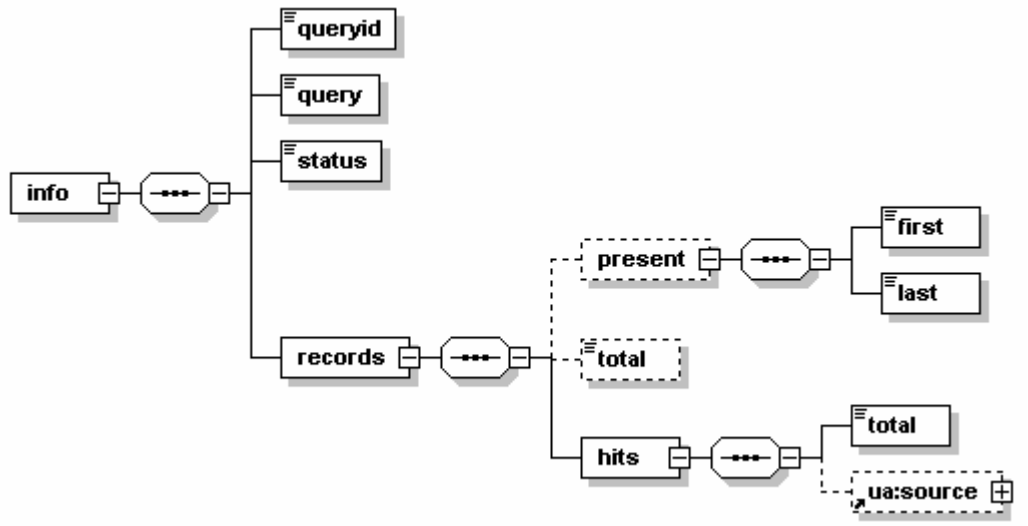



Figura 4.29 – Diagrama do elemento “info”.

O elemento “present”, opcional, e também pertencente ao elemento “records”, permite especificar o intervalo de registos presente no próprio documento. Daí que inclua ainda dois outros elementos, “first” e “last”, para especificar os limites desse intervalo.

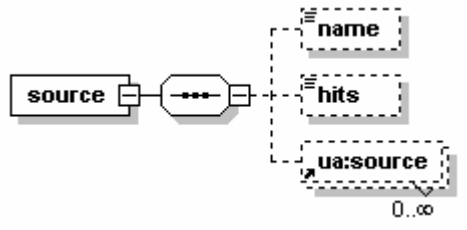



Figura 4.30 – Diagrama do elemento “source”.

O elemento “source”, pertencente ao elemento “hits”, encontra-se representado em diagrama na Figura 4.30. Este elemento possui o mesmo nome e até uma semântica idêntica ao elemento “dc:source”. Pois o objectivo de ambos é identificar possíveis fontes à qual um recurso pertence. Contudo, e neste contexto, o elemento “ua:source” vai mais longe ao ser definido como um elemento recursivo e ser possuidor de mais informação que a que está prevista para o “dc:source”.

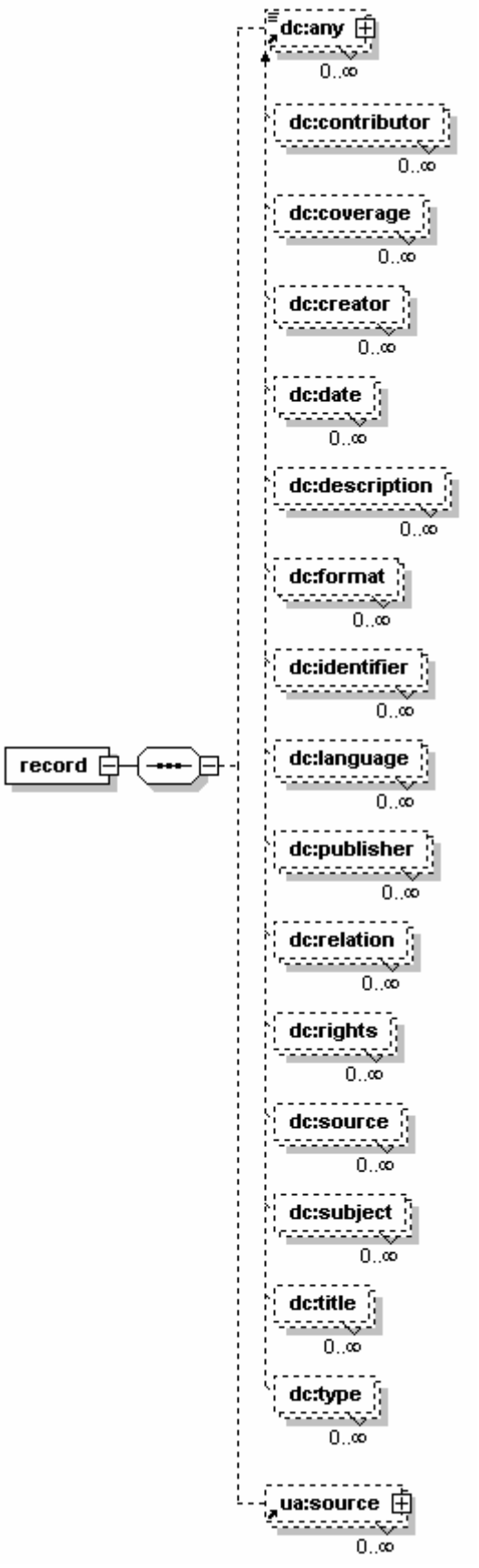



Figura 4.31 – Diagrama do elemento “record”.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:results xmlns:ua="http://www.ua.pt/spri/results"
            xmlns:dc="http://purl.org/dc/elements/1.1/">
 <ua:info>
  <ua:queryid>query1069980348339</ua:queryid>
  <ua:query>Title: java</ua:query>
  <ua:status>terminated</ua:status>
  <ua:records>
   <ua:present>
    <ua:first>1</ua:first>
    <ua:last>10</ua:last>
   </ua:present>
   <ua:total>50</ua:total>
   <ua:hits>
    <ua:total>233</ua:total>
    <ua:source uri="urn:spri:main">
     <ua:source uri="urn:ua:opacs">
      <ua:source uri="urn:opacs:aberdeen">
       <ua:hits>100</ua:hits>
      </ua:source>
      <ua:source uri="urn:opacs:austin">
       <ua:hits>133</ua:hits>
      </ua:source>
     </ua:source>
    </ua:source>
   </ua:hits>
  </ua:records>
 </ua:info>
 <ua:record id="1">
  <dc:title>Java in a nutshell : a desktop quick reference.</dc:title>
  <dc:creator>Flanagan, David.</dc:creator>
  <dc:type>text</dc:type>
  <dc:publisher>Sebastopol : O'Reilly,</dc:publisher>
  <dc:date>2002.</dc:date>
  <dc:language>eng</dc:language>
  <dc:identifier>ISBN:0596002831</dc:identifier>
  <ua:source uri="urn:spri:main">
   <ua:source uri="urn:ua:opacs">
    <ua:source uri="urn:opacs:aberdeen">
     <ua:name>Aberdeen University</ua:name>
    </ua:source>
    <ua:source uri="urn:opacs:austin">
     <ua:name>Austin College</ua:name>
    </ua:source>
   </ua:source>
  </ua:source>
 </ua:record>
 ...
</ua:results>
```

Figura 4.32 – Exemplo de um documento resposta com “info” e “record”.

O elemento “source”, representado na Figura 4.30, poderá possuir: um elemento “name”, onde é possível colocar o nome da fonte; um elemento “hits”, onde colocar o número de registos encontrados nessa fonte; e múltiplos elementos “source”, iguais em estrutura ao elemento “source” aqui descrito. Mais uma vez presente o conceito de recursividade no modelo de dados, permitindo a pormenorização da informação até à exaustão. Este elemento possui ainda o atributo “uri”, não representado no diagrama, que permite a identificação da fonte.

O elemento “record” aparece sempre nos documentos de resposta dos métodos submitQuery(), getRecords() e getRecord() e é aquele que transporta a informação dos registos individuais. O seu diagrama de composição encontra-se representado na Figura 4.31. Este elemento é composto por elementos do modelo de metadados Dublin Core, que poderão aparecer em qualquer posição e em qualquer número, e pelo elemento “source”, já descrito anteriormente. Possui também um atributo “id”, obrigatório, que permite a identificação unívoca dentro do conjunto de registos recolhidos pela pesquisa.

Na Figura 4.32 encontra-se um exemplo de um documento resposta em que se combinam os elementos “info” e “record” sob o elemento “results”.

#### 4.6.2.3 O elemento sources

O elemento “sources” não pode aparecer em conjunto com qualquer dos outros elementos também pertencentes ao elemento “results”. Este elemento permite criar um documento de resposta para informar acerca das fontes que poderão ser explicitamente pesquisadas. É, por isso, o resultado da chamada do método getSources(). No diagrama da Figura 4.33, encontra-se representada a sua composição.

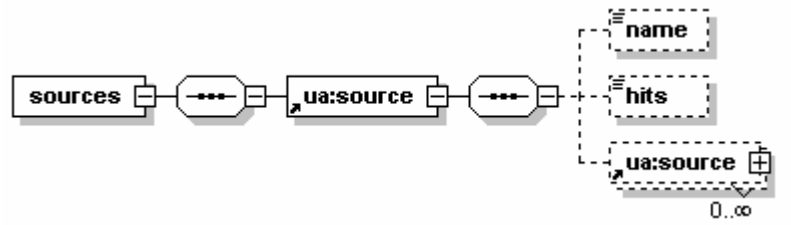



Figura 4.33 – Diagrama do elemento “sources”.

Este elemento é composto apenas por um elemento “source”, o qual já foi descrito anteriormente. Com esta estrutura de elementos, é possível descrever toda a estrutura hierárquica de fontes de pesquisa.

Neste caso, a utilização do elemento “source”, sem o elemento “sources”, seria suficiente, contudo pretende-se garantir, sem ambiguidades, que este documento consiste de facto numa resposta a um pedido de enumeração de fontes de pesquisa.

Na Figura 4.34 encontra-se ilustrado um exemplo deste tipo de documento.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:results xmlns:ua="http://www.ua.pt/spri/results">
 <ua:sources>
  <ua:source uri="urn:spri:main">
   <ua:source uri="urn:ua:archive">
    <ua:name>Arquivo da Universidade de Aveiro</ua:name>
   </ua:source>
   <ua:source uri="urn:ar:archive">
    <ua:name>Arquivo da Assembleia da República</ua:name>
   </ua:source>
   <ua:source uri="urn:pj:archive">
    <ua:name>Arquivo da Provedoria de Justiça</ua:name>
   </ua:source>
   <ua:source uri="urn:ua:opacs">
    <ua:name>Catálogo Colectivo na UA</ua:name>
    <ua:source uri="urn:opacs:aberdeen">
     <ua:name>Aberdeen University</ua:name>
    </ua:source>
    <ua:source uri="urn:opacs:austin">
     <ua:name>Austin College</ua:name>
    </ua:source>
    <ua:source uri="urn:opacs:abell">
     <ua:name>Abell Library Center Austin College</ua:name>
    </ua:source>
   </ua:source>
  </ua:source>
 </ua:sources>
</ua:results>
```

Figura 4.34 – Exemplo de um documento com o elemento “sources”.

#### 4.6.2.4 O elemento indexes

O elemento “indexes” também não pode aparecer em simultâneo com qualquer dos outros elementos pertencentes ao elemento “results”. Este elemento tem por objectivo criar um documento de resposta para um pedido de índices, ou seja, o resultado da chamada do método getIndex(). A sua composição encontra-se representada na Figura 4.35.


Figura 4.35 – Diagrama do elemento “indexes”.

Este elemento é composto pelo elemento “index”, que pode ser repetido múltiplas vezes e pode conter uma lista de valores, utilizando o elemento “item”. O elemento “index” possui ainda o atributo “name”, não representado no diagrama, que permite identificar o nome do índice.

Também neste caso, o elemento “index” poderia aparecer no documento sem ter de pertencer a um elemento “indexes”, contudo, mais uma vez se pretende aqui garantir, sem ambiguidades, que o documento resposta é de facto uma resposta a um pedido de índices.

Na Figura 4.36 encontra-se representado um exemplo da utilização destes elementos para devolver listas de valores para alguns índices.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:results xmlns:ua="http://www.ua.pt/spri/results">
 <ua:indexes>
  <ua:index name="dc:author">
   <ua:item>Alberto</ua:item>
   <ua:item>Bruno</ua:item>
   ...
   <ua:item>Zagalo</ua:item>
  </ua:index>
  <ua:index name="dc:date">
   <ua:item>2004-07</ua:item>
   <ua:item>2003-05</ua:item>
   ...
   <ua:item>2000-11</ua:item>
  </ua:index>
  <ua:index name="dc:subject">
   <ua:item>Programação</ua:item>
   <ua:item>Sistemas</ua:item>
   ...
   <ua:item>XML</ua:item>
  </ua:index>
 </ua:indexes>
</ua:results>
```

Figura 4.36 – Exemplo de um documento com o elemento “indexes”.

## 4.7 Revisão

Este capítulo apresenta a proposta de um modelo e de uma arquitectura para uma plataforma de *middleware* que ofereça suporte à criação de bibliotecas digitais com repositórios de informação distribuídos e heterogéneos. Podendo estes últimos, serem já eles próprios bibliotecas digitais na sua plenitude.

São apresentados os modelos funcionais e de dados para a arquitectura, partindo de um denominador comum: a tecnologia XML. Esta tecnologia, aparece nesta proposta, como uma espécie de panaceia para todos os problemas. Na verdade, a sua utilização é actualmente tida como um meio inquestionável para a resolução de muitos problemas de interoperabilidade. Daí que o seu uso intensivo nesta arquitectura deve ser visto como uma tentativa para atingir um elevado nível de interoperabilidade: tanto ao nível da transferência de dados como ao nível das interfaces funcionais.

Esta arquitectura assenta sobre alguns conceitos de base conhecidos, como a distribuição e o paralelismo, já aplicados anteriormente com sucesso por outros investigadores (Lagoze and Davis, 1995a; Lagoze et al., 1995b). Contudo pretende apresentar-se de forma mais original com base no conceito de uma modelação conceptual recursiva, ao nível da funcionalidade e nalguns pontos ao nível dos dados, o que lhe permite apresentar-se de uma forma minimalista, mantendo a possibilidade de dar origem a sistemas de um elevado grau de complexidade.

O elevado nível de interoperabilidade que esta arquitectura tenta atingir será, porventura, a sua maior contribuição para a evolução das arquitecturas das bibliotecas digitais futuras. Este elevado nível de interoperabilidade leva, de forma completamente natural, a uma enorme escalabilidade e, sobretudo, a uma extrema facilidade de adaptação e integração de novos e estranhos sistemas na plataforma.

A plataforma de *middleware* aqui apresentada possui a capacidade inata de criar uma federação de bibliotecas digitais na qual são sempre bem vindos novos membros – novos repositórios de informação ou novas bibliotecas digitais – devido à facilidade com que podem ser integrados na federação. Isto, independentemente do tipo de sistema em que se encontram. Cada repositório ou biblioteca digital é representada nesta plataforma através de um elemento funcional integrador, o qual também pode ser chamado de agregador visto poder contribuir para a recolha de informação a partir de múltiplas fontes.

# Capítulo 5

# Caso de Estudo: Agregador de Registos Bibliográficos

## 5.1 Introdução

No presente capítulo, é descrita a arquitectura e desenvolvimento de um agregador de registos bibliográficos. Sistema que foi concebido e implementado com o objectivo de ser integrado no sistema de recolha bibliográfica do projecto Memória de África (MemAfrica, 2009), a partir do conceito de catálogo colectivo virtual.

Em virtude da funcionalidade oferecida e do tipo de operações que tem de efectuar, este sistema consiste num demonstrador efectivo das potencialidades do SPRI - Serviço de Pesquisa e Recolha de Informação, descrito no capítulo anterior. Ou seja, um serviço com a capacidade de consultar múltiplas e diversas fontes de informação, simultaneamente, e apresentar os resultados como um conjunto de dados unificado e coerente.

Assim e com base nos conceitos descritos anteriormente, do modelo de abstracção e da arquitectura da plataforma de *middleware*, este Agregador de Registos Bibliográficos é tido como uma instanciação breve dessa plataforma e pese embora a sua escala mais reduzida e a utilização de algumas tecnologias diferentes, este sistema é aqui utilizado

para validar as ideias de concepção da plataforma de *middleware* e fornecer potenciais indicadores sobre o comportamento e desempenho desta última.

## 5.2 Requisitos

Enumeram-se de seguida os diversos requisitos funcionais que se colocam ao Agregador de Registos Bibliográficos:

- a pesquisa simultânea em múltiplos servidores Z39.50;
- a criação de conjuntos unificados de registos, em resposta aos pedidos de pesquisa;
- a identificação e remoção de registos duplicados;
- a normalização do formato dos registos recebidos, assim como a adopção de formatos simplificados para essa normalização;
- a apresentação de uma interface de utilizador, completamente baseada na web.

O último requisito, da apresentação da informação ao utilizador, não se encontra contemplado nos requisitos da plataforma de *middleware*, pelas razões óbvias. Contudo, pretendeu-se, na implementação deste sistema, colocar de imediato à prova a sua capacidade de comunicação com os seus utilizadores finais. O que não invalida, a sua plena integração numa plataforma de pesquisa mais vasta, como é a plataforma descrita no capítulo anterior.

## 5.3 A Arquitectura

A arquitectura do Agregador de Registos Bibliográficos assenta sobre os conceitos já descritos no capítulo anterior, na arquitectura da plataforma de *middleware*. Aqui são materializados os conceitos de pesquisa distribuída e paralela e, grosso modo, os módulos do elemento funcional SPRI.

A arquitectura deste sistema encontra-se representada na Figura 5.1 e é constituída basicamente por dois módulos:

- o SPD - Sistema de Pesquisas Distribuídas – responsável pelas pesquisas aos servidores Z39.50 distribuídos na rede e por todas as operações sobre as respostas recebidas desses mesmos servidores; engloba os módulos Conversor DC e Cliente da arquitectura do SPRI;

- e o WS - *Web Service* – responsável por fornecer uma interface *web service* com o exterior e corresponde ao módulo, com o mesmo nome, na arquitectura do SPRI.

A Figura 5.1 sugere também que estes módulos devem ser implementados utilizando a plataforma JAVA e que as entradas e saídas do sistema são documentos XML.

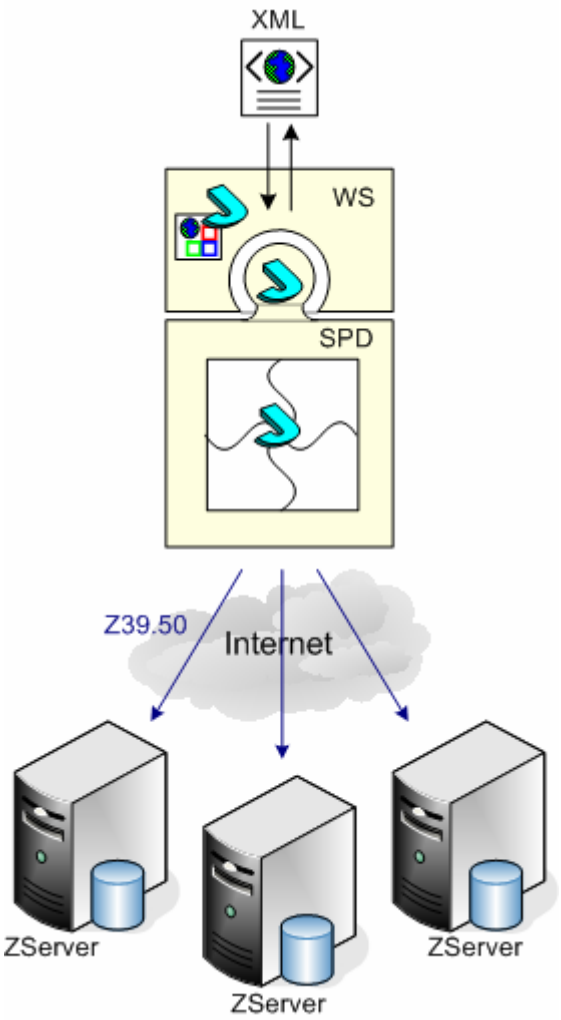



Figura 5.1 – Arquitectura do Agregador de Registos Bibliográficos.

### 5.3.1 O Módulo SPD

A Figura 5.1 sugere que o módulo SPD seja constituído por um grupo de subelementos ligados entre si. Na Figura 5.2, encontra-se representada em detalhe a composição e arquitectura do módulo SPD.

A arquitectura do módulo SPD segue uma filosofia orientada ao componente. O que significa que o módulo em si é composto por um conjunto de componentes

independentes que podem ser usados individualmente, para cumprir tarefas específicas, ou podem ser reorganizados por forma a implementarem um módulo com outro objectivo. De notar também que alguns componentes encontram-se agrupados num subconjunto dentro do módulo, chamado subconjunto interno de componentes. Este subconjunto encontra-se assim esquematicamente agrupado por possuir uma relação mais próxima entre os seus componentes.

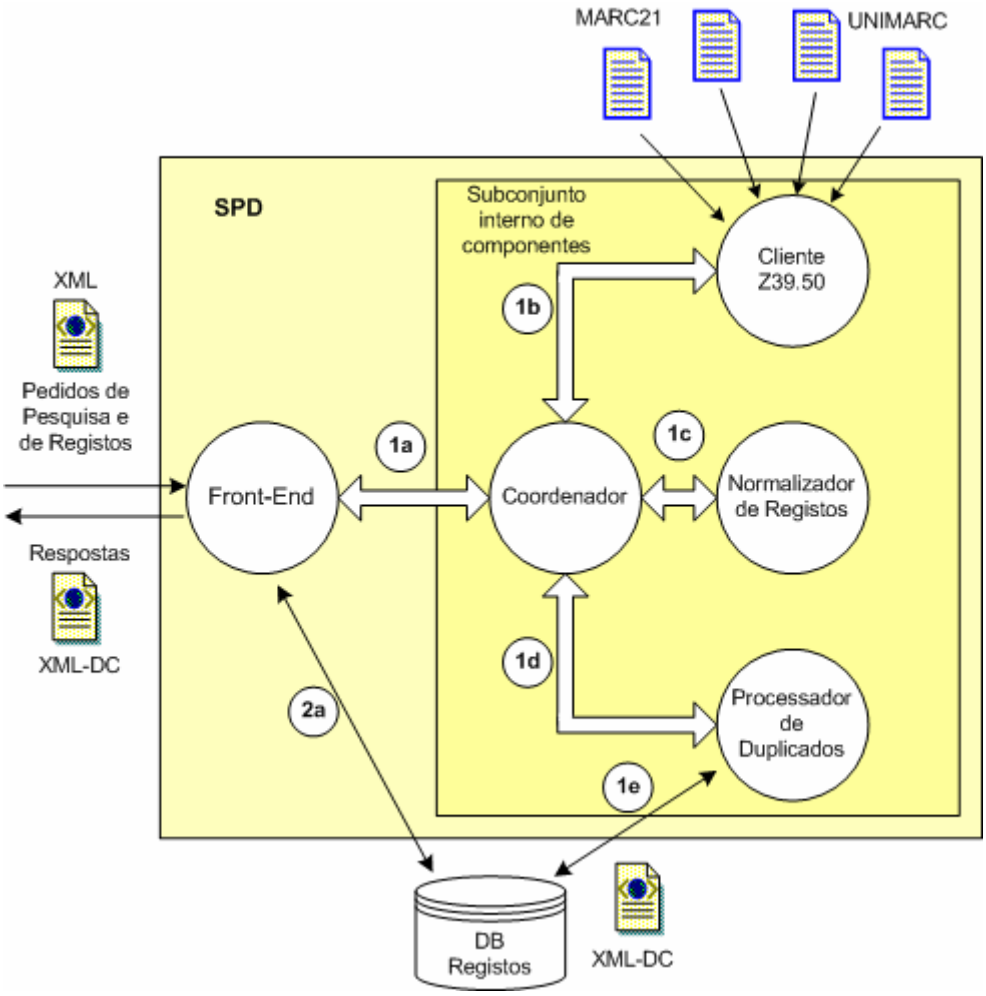



Figura 5.2 – Arquitectura do módulo SPD.

Os principais componentes que constituem o módulo SPD, são:

- o “Cliente Z39.50”;
- o “Normalizador de Registos”;
- o “Processador de Duplicados”;
- o “Coordenador”;
- o “Front-End”.

O componente “Cliente Z39.50” implementa o módulo Cliente da arquitectura do SPRI. Já os componentes “Normalizador de Registos” e “Processador de Duplicados” implementam o módulo Conversor DC. Os restantes componentes, são componentes de gestão interna e interface do módulo SPD.

Na Figura 5.2 encontram-se também ilustradas as duas linhas principais de execução do módulo, que representam duas operações genéricas:

- (1) – a execução de uma pesquisa;
- (2) – a execução de um pedido de registos.

As diversas fases, pelas quais estas linhas passam, estão numeradas para uma melhor compreensão da sua ordem de execução.

Explicando as diversas fases das linhas de execução:

- (1a) – o pedido de pesquisa acabou de ser entregue ao componente “Front-End”, que o entrega de imediato ao componente “Coordenador”;
- (1b) – o pedido de pesquisa é entregue ao componente “Cliente Z39.50”, que devolve a resposta (na forma de registos) assim que esta estiver pronta;
- (1c) – a resposta devolvida pelo “Cliente Z39.50”, é entregue ao componente Normalizador de Registos que a devolve assim que terminar o seu processamento;
- (1d) – a resposta, depois de normalizada, é entregue ao componente “Processador de Duplicados”, que constrói um conjunto de registos unificado, como resposta da pesquisa;
- (1e) – o componente “Processador de Duplicados” salvaguarda o conjunto de registos numa base de dados, onde residirá temporariamente para futuros pedidos de registos;
- (2a) – o pedido de registos acabou de ser entregue ao componente “Front-End”, que executa de imediato uma operação de busca à base de dados; se o conjunto de registos pedidos existir, este é devolvido, caso contrário será devolvida uma mensagem de erro.

De notar que a segunda operação – pedido de registos – deve sempre ser feita no contexto de uma primeira operação – pedido de pesquisa – bem sucedida. Se tal não for o caso, será devolvida uma mensagem de erro pela operação. A cada um dos componentes está atribuída uma funcionalidade bem definida e encerram em si mesmos a tentativa de resolução de alguns dos problemas mais emblemáticos no domínio da

investigação sobre catálogos colectivos virtuais. Segue-se por isso, uma explanação sobre esses componentes.

#### 5.3.1.1 O Componente "Cliente Z39.50"

O componente "Cliente Z39.50" materializa o desafio de engenharia de permitir a pesquisa paralela de diversos servidores, sobre o protocolo Z39.50, que, como se sabe, permite apenas ligações ponto a ponto. Este componente, poder-se-á dizer, constitui o cerne do módulo. Implementa todo o trabalho de distribuição dos pedidos de pesquisa pelos múltiplos servidores Z39.50 e de recolha dos registos que estes enviam em resposta.

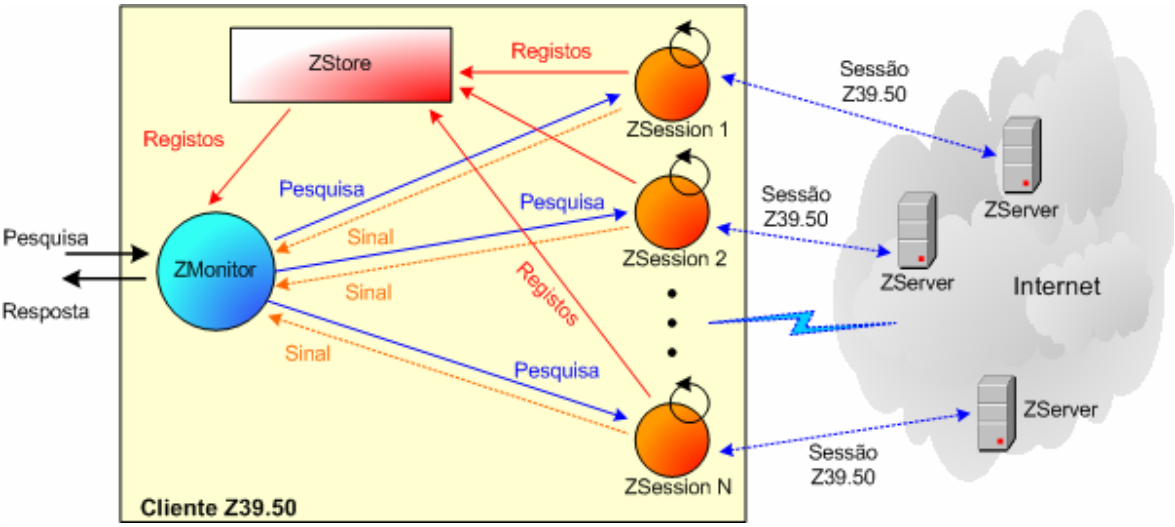



Figura 5.3 – Arquitectura do componente "Cliente Z39.50".

Na Figura 5.3 encontra-se representada a arquitectura deste componente, onde se podem ver os vários objectos que o compõem e os fluxos de dados e de sinalização entre os mesmos. Os objectos são:

- ZSession – responsável pela implementação do protocolo Z39.50 e por isso da criação e manutenção de uma sessão com um servidor Z39.50, assim como do envio do pedido de pesquisa ao servidor a que se encontra ligado e da recepção dos registos que este envia;
- ZMonitor – responsável por monitorizar os diversos objectos ZSession na execução das pesquisas;

- ZStore – permite a guarda e agrupamento dos registos que chegam de todas as sessões em simultâneo.

Por forma a ficar-se com uma ideia mais clara da relação entre estes objectos e o seu funcionamento segue-se a explicação da execução de um pedido de pesquisa colocado ao componente.

O pedido de pesquisa chega ao componente através do objecto ZMonitor. Este cria as devidas sessões (objectos ZSession) com os servidores Z39.50 e passa-lhes o pedido de pesquisa. Cada objecto ZSession envia então o pedido ao servidor a que se encontra ligado e aguarda pela chegada da sua resposta. A resposta do servidor começa por indicar a quantidade de registos encontrados para a pesquisa em questão e cabe depois ao objecto ZSession pedir a quantidade de registos que pretende. À medida que os registos vão chegando ao objecto ZSession, estes vão sendo enviados para o objecto ZStore que se encarrega de os agrupar com os restantes registos recebidos pelos demais objectos ZSession.

Em simultâneo com este processo, o objecto ZMonitor procede a duas tarefas simultaneamente: vai acedendo ao objecto ZStore, retirando os registos, um a um, e passando-os ao objecto exterior, seu cliente; monitoriza também o estado de execução em que se encontram os objectos ZSession e quando estes terminarem a recepção de registos, então dá por encerrada a tarefa de pesquisa.

Como é patente na figura, através da ilustração de um círculo contendo um apontador, cada objecto ZSession possui uma linha de execução própria (*thread*), daí a possibilidade de todas as pesquisas se realizarem em paralelo.

Este componente pode ser usado de forma independente, integrado noutros sistemas, para servir de motor de busca sobre o protocolo Z39.50, como por exemplo: sistemas de catalogação para bibliotecas ou aplicações proprietárias para citação bibliográfica.

#### 5.3.1.2 O Componente “Normalizador de Registos”

Os registos bibliográficos, recebidos pelo componente “Cliente Z39.50”, possuem uma multitude de formatos, por vezes bastante complexos, como são exemplo todas as nuances do formato MARC. A normalização destes registos é uma operação que se impõe no actual sistema, tendo por base os requisitos apresentados anteriormente.

Em primeiro lugar, há a necessidade de identificar e remover registos que referenciam a mesma obra. Esta identificação passa por uma tarefa de comparação dos próprios registos. Ora tal comparação só é possível entre objectos comparáveis, ou seja,

possuindo a mesma estrutura. Embora não impossível, a comparação de registos nos seus formatos originais seria uma tarefa dantesca em termos de complexidade e não traria qualquer utilidade em termos finais.

Em segundo lugar, pretende-se que os registos assumam um formato mais simples. Este requisito vai ao encontro de uma maior acessibilidade por parte dos utilizadores à informação. Para que isto seja possível, tem de se evitar que os registos que chegam aos utilizadores sejam demasiado complexos e exijam conhecimento especializado sobre os mesmos.

Desta forma, o componente "Normalizador de Registos" tem por missão a uniformização do formato de todos os registos recebidos para um formato suficientemente simples e compreensível para a maioria dos utilizadores.

#### 5.3.1.3 O Componente "Processador de Duplicados"

Considerando o facto de que a maior parte das bibliotecas possuem cópias das mesmas obras, então é lógico que quando é feita uma mesma pesquisa nessas bibliotecas surjam registos, oriundos de diferentes bibliotecas, que referenciam a mesma obra.

O componente "Processador de Duplicados", presente nesta arquitectura, possui precisamente a missão de tentar identificar e remover registos com o mesmo significado, salvaguardando numa única cópia do registo todas as proveniências do mesmo. Cada registo aceite, ou actualizado, por este componente é enviado a uma base de dados, onde é armazenado para posterior recolha e consulta.

#### 5.3.1.4 O Componente "Coordenador"

Como o próprio nome indica, este componente tem por missão a coordenação dos diversos componentes descritos anteriormente. Por outras palavras, é este o componente que comanda a linha de execução que atravessa os diversos componentes pertencentes ao subconjunto interno de componentes. Quando na secção 5.3.1.1 se diz que o objecto ZMonitor entrega os registos, um a um, ao seu cliente, de facto neste caso, está a entregá-los ao componente "Coordenador", que depois os faz passar pelo resto do circuito – componente "Normalizador de Registos" e componente "Processador de Duplicados".

#### 5.3.1.5 O Componente “Front-End”

O componente “Front-End“ é o módulo que implementa a interface com o exterior. É este o componente responsável por todas as interacções de entrada e saída do módulo – aceita pedidos de pesquisa, ou de registos, e devolve os resultados obtidos.

Quando se tratam de pedidos de pesquisa, estes são encaminhados para o componente “Coordenador”. Quando são pedidos de registos, então este próprio componente, pesquisa a base de dados à sua procura. Este processo afigura-se mais eficiente que entregar todo o trabalho ao componente “Coordenador”, o que em última instância faria com que a existência do próprio componente “Front-End” fosse posta em causa.

### 5.3.2 O Módulo WS

O módulo WS - *Web Service* encontra-se acoplado ao módulo SPD e tem como principal missão oferecer um meio de acesso distribuído ao Agregador de Registos Bibliográficos, baseado nos *web services*.

O Agregador pode ser perspectivado como uma espécie de *proxy* entre clientes e servidores Z39.50 e, por isso, um meio de aceder a esse tipo de servidores sem o conhecimento específico do protocolo.

Assim, o módulo WS disponibiliza um meio de acesso distribuído a servidores Z39.50, mas com uma interface muito mais simples, mais leve e sem as limitações de acesso impostas muitas vezes pelos *firewalls*. O que fomenta a interoperabilidade com este tipo de servidores.

O aparecimento dos protocolos SRU/SRW veio precisamente ao encontro da ideia do aumento da interoperabilidade entre sistemas bibliográficos, propondo a substituição dos tradicionais servidores Z39.50 por servidores web com idêntica funcionalidade. Contudo, estes protocolos continuam a manter um elevado nível de complexidade ao permanecerem muito fieis ao modelo de dados do protocolo Z39.50.

Desta forma, o módulo WS oferece um serviço, que vai ao encontro das novas tendências no domínio do sistemas bibliográficos, mas não pretende sugerir qualquer substituição dos protocolos já existentes, até porque se pretende oferecer uma interface funcional de muito mais alto nível e muito mais simples.

A solução proposta no presente trabalho vai no sentido do aproveitamento das soluções já estabelecidas. Aproveitando o largo parque de servidores Z39.50 existente, é possível adicionar mais valias e contornar os eventuais problemas que esse protocolo,

por vezes, levanta. A substituição integral desses sistemas, coloca à partida grandes desafios, de ordem técnica e financeira, que a maioria das instituições não está preparada para enfrentar.

## 5.4 O Desenvolvimento

Nesta secção serão abordadas as soluções técnicas adoptadas para o desenvolvimento dos dois módulos: SPD e WS.

Para o desenvolvimento dos dois módulos, foi escolhida a plataforma JAVA. As suas capacidades nativas para o desenvolvimento de aplicações concorrentes e de rede, assim como a sua portabilidade e possibilidades para o desenvolvimento de componentes foram determinantes na a sua escolha. Todos os componentes do módulo SPD foram desenvolvidos como JavaBeans (Hamilton, 1997), para permitirem a sua fácil integração em meios diversos. Este é precisamente o significado da imagem do módulo SPD, na Figura 5.1, que aparece como um conjunto de peças de um puzzle encimado pela letra J.

### 5.4.1 O Módulo SPD

#### 5.4.1.1 Paradigma Funcional

O paradigma funcional adoptado, e que norteou a implementação do módulo SPD, é um paradigma orientado ao pedido, significando que os pedidos dos utilizadores possuem precedência sobre os recursos a utilizar. É assunção feita por este paradigma de que, actualmente, as capacidades de escalabilidade dos sistemas devem tentar sempre superar os requisitos impostos sobre os mesmos.

Neste paradigma, o componente “Front-End” detém o papel de atender todos os pedidos que lhe chegam do exterior e, para cada um, proceder a uma nova instanciação do subconjunto interno de componentes. A cada pedido são associados dois parâmetros: tempo máximo de pesquisa (*timeout*) e número mínimo de registos para satisfação do pedido. Quando um destes parâmetros é atingido, os resultados obtidos são de imediato devolvidos ao utilizador. Caso o pedido tenha sido satisfeito, mas existam ainda registos para recolher, então o subconjunto interno de componentes continua a trabalhar na pesquisa por sua conta até ao término da operação, não colidindo com outros pedidos de pesquisa que sejam submetidos ao módulo, entretanto. Quando a operação de pesquisa

termina, todo o subconjunto de componentes, previamente instanciado para a pesquisa, é libertado e o espaço que ocupava fica a cargo do mecanismo de *garbage collector* da JVM - *Java Virtual Machine*.

Este modelo de funcionamento obriga, por outro lado, a que, para cada pedido de pesquisa, sejam instanciadas novas ligações com os servidores Z39.50. Sendo abertas no início dos pedidos e encerradas imediatamente no fim da sua satisfação. Este comportamento não prevê um grande aproveitamento das capacidades de sessão do protocolo Z39.50. Contudo, possui a vantagem de simplificar ao máximo a gestão dessas ligações. O aproveitamento de ligações, já estabelecidas, para diversos pedidos de pesquisa, levaria a um elevado grau de complexidade na sua gestão, até porque muitos servidores não suportam ainda a capacidade de operações concorrentes sobre uma mesma ligação.

#### 5.4.1.2 O Componente “Cliente Z39.50”

Como descrito na arquitectura, o componente “Cliente Z39.50” é o responsável pela implementação de sessões sobre o protocolo Z39.50. Com vista a obter alguma ajuda na implementação deste componente, foi adquirido um produto comercial, o ZedJAVA (Crossnet, 2001). Este produto consiste numa biblioteca de classes desenvolvida em JAVA que fornece funcionalidades para codificar e descodificar mensagens conformes ao protocolo Z39.50. No momento da compra deste produto, não existiam propostas, comerciais ou outras, de produtos para o desenvolvimento da parte dinâmica do protocolo. Desta forma, foi usado o ZedJAVA para a implementação estática do protocolo, tendo a parte dinâmica sido inteiramente desenvolvida.

A parte dinâmica do protocolo Z39.50 encontra-se totalmente implementada no objecto ZSession, que foi desenvolvido em total conformidade com as tabelas de estado do protocolo. A criação e interpretação das mensagens (PDU - *Protocol Data Units*) foi implementada num objecto complementar ao ZSession, de nome ZAssociation, que faz uso das facilidades da API ZedJAVA.

#### 5.4.1.3 O Componente “Normalizador de Registos”

Conforme descrito na arquitectura, o componente “Normalizador de Registos” tem por missão a uniformização e simplificação dos registos recebidos pelo sistema.

Aqui, mais uma vez, foi seguida a recomendação da arquitectura da plataforma de *middleware*, adoptando o formato Dublin Core simples para a normalização da

informação. Desta forma, evita-se o posterior desenvolvimento de um módulo específico para a normalização para esse formato, para integração na plataforma.

Na Figura 5.4 encontra-se um exemplo de um registo convertido para Dublin Core. Este registo obedece inteiramente ao modelo de dados das respostas, descrito no capítulo anterior.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ua:record xmlns:ua="http://www.ua.pt/spri/results"
           xmlns:dc="http://purl.org/dc/elements/1.1/" id="12">
 <dc:title>AXIS : the next generation of Java SOAP /</dc:title>
 <dc:creator>Basha, S. Jeelani.</dc:creator>
 <dc:creator>Irani, Romin.</dc:creator>
 <dc:type>text</dc:type>
 <dc:publisher>Birmingham : Wrox Press,</dc:publisher>
 <dc:date>2002.</dc:date>
 <dc:language>eng</dc:language>
 <dc:identifier>ISBN:1861007159</dc:identifier>
 <ua:source uri="urn:opacs:abell">
  <ua:name>Abell Library Center Austin College</ua:name>
 </ua:source>
</ua:record>
```

Figura 5.4 – Registo convertido para Dublin Core simples.

### *O processo de normalização*

Para que um registo tome a forma do exemplo da Figura 5.4, são necessários vários passos de conversão. Considerando, por exemplo, que o formato de origem é o formato MARC, em qualquer das suas nuances, então são necessários dois tipos de conversão:

- do formato de transporte ISO2709 para XML;
- do formato MARC para Dublin Core simples.

A primeira conversão é concretizada com a ajuda do MARCXML Toolkit, uma API JAVA, também disponível na LoC e desenvolvida pela mesma no âmbito da sua iniciativa MARCXML (LoC, 2009f), que promove a disponibilização do formato MARC sobre XML.

A segunda conversão é efectuada com recurso a XSLTs (W3C, 1999c). Existe um XSLT, pertencente à LoC, desenvolvido no âmbito da mesma iniciativa, que efectua a conversão de MARC21 (LoC, 2006) para Dublin Core e é parcialmente visível na Figura 5.5.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:marc="http://www.loc.gov/MARC21/slim"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ua="http://www.ua.pt/spri/results"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes="marc">
<xsl:import href="MARC21slimUtils.xsl"/>
<xsl:output method="xml" version="1.0" encoding="iso-8859-1" indent="yes"/>

<xsl:template match="/">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="marc:record">
  <xsl:variable name="leader" select="marc:leader"/>
  <xsl:variable name="leader6" select="substring($leader,7,1)"/>
  <xsl:variable name="leader7" select="substring($leader,8,1)"/>
  <xsl:variable name="controlField008" select="marc:controlfield[@tag=008]"/>
...
```

Figura 5.5 – XSLT de conversão XML/MARC para XML/Dublin Core (LoC, 2009f).

Este XSLT foi contudo sujeito a algumas alterações por forma a que o registo pudesse ser albergado no elemento “ua:record” e para que o elemento “dc:identifier” albergasse correctamente o ISBN - *International Standard Book Number* ou o ISSN - *International Standard Serial Number* da obra.

O XSLT de conversão a partir de Unimarc, foi desenvolvido no âmbito deste trabalho, a partir de um documento de trabalho da UKOLN. É praticamente igual ao XSLT de conversão a partir de MARC21, ressalvando o facto de os números que identificam os campos serem diferentes.

Este sistema de uso de XSLTs para efectuar a conversão, garante ao componente a expansibilidade da sua capacidade de conversão. Para que uma nova conversão seja suportada, apenas é necessário fornecer o XSLT adequado.

#### 5.4.1.4 O Componente “Processador de Duplicados”

A identificação de registos duplicados é uma tarefa à qual tem sido dispensada larga atenção por parte de múltiplos grupos de investigação. Aparentemente, esta tarefa poderia ser muito simples, resumindo-se à verificação da igualdade de todos os campos dos registos. Contudo, o processo está muito longe de ser tão simples. Ao fazermos uma simples pesquisa em diversas bibliotecas, facilmente deparamos com registos que referenciam a mesma obra, mas possuem, por vezes, diferenças substanciais nos seus

campos. Isto acontece devido às múltiplas catalogações a que uma obra está sujeita dentro das múltiplas bibliotecas onde se encontra. Por isso, este problema afigura-se mais complexo que aquilo que inicialmente se supõe.

Não é objectivo deste trabalho, fazer uma investigação exaustiva da forma de reconhecer registos duplicados, contudo foi concebido e desenvolvido um método no presente trabalho, por forma a oferecer mais uma solução, embora muito particular, para este problema.

Nos últimos anos, alguns grupos de investigação a trabalhar neste domínio conceberam, com algum grau de sucesso, vários métodos para a identificação de registos duplicados. Esses métodos são bastante diferenciadas no tocante ao poder de processamento exigido: desde a simples comparação de um identificador até ao reconhecimento por amostragem.

No presente trabalho, a norma seguida tem sido a adopção de soluções simples. Também, nesta situação particular é de interesse que o processamento exigido seja baixo, por forma a aligeirar o impacto dos múltiplos componentes em execução simultânea. Desta forma, os métodos de identificação de registos duplicados utilizados são apenas dois:

- Identificação através do campo identificador (dc:identifier) – este método baseia-se na comparação do ISBN ou ISSN, quando presentes; quando não se encontram presentes, todo o conteúdo do campo é comparado;
- Identificação através de vários campos do registo – este método baseia-se na comparação de diversos campos do registo, que se crêem ser vitais para a identificação da obra que referenciam. Esses campos são: o campo título (dc:title); os campos de autoria (dc:creator), tanto em conteúdo como em número; e caso existam, mas não obrigatório, os campos editor (dc:publisher), tipo (dc:type), linguagem (dc:lang) e data (dc:date).

No primeiro método, caso o campo identificador (dc:identifier) exista, é necessário proceder à extracção do ISBN ou ISSN, caso estes existam. A necessidade desta tarefa decorre do facto de no processo de conversão, no componente “Normalizador de Registos”, ser por vezes adicionada informação a esse campo que não faz parte do conjunto de caracteres do ISBN ou ISSN.

O segundo método foi concebido no âmbito deste trabalho e é mais exigente em termos de poder de processamento, contudo é utilizado apenas quando o primeiro método não o pode ser, devido à inexistência do campo identificador. Neste método a

comparação de campos cinge-se aos campos que podem colocar em causa a similaridade dos registos. Os restantes campos do Dublin Core como, por exemplo, o dc:subject, o dc:description ou o dc:rights não são tidos em conta por se pensar que o seu conteúdo não contribui de forma decisiva para a identificação unívoca dos registos.

Estes dois métodos estão muito longe de serem os melhores para a execução da tarefa de identificação de duplicados, contudo pretendeu-se chegar a um compromisso entre a fiabilidade e a "leveza" do processo.

Estes métodos de identificação de registos duplicados, são aplicados a todos os registos que passam por este componente. Após esta fase, um registo não identificado como duplicado é simplesmente adicionado à base de dados, enquanto um registo identificado como tal é descartado e a sua origem é adicionada ao registo idêntico que já se encontra na base de dados.

#### 5.4.1.5 O Componente "Front-End"

O componente "Front-End", como responsável pelas interacções com o exterior deve apresentar uma interface programática capaz de satisfazer todos os pedidos de informação passível de ser requisitada a um sistema com estas características.

Com esta ideia em mente, foi decidido adoptar a interface funcional comum proposta no capítulo anterior. Nessa interface, nem todos os métodos encontrarão a sua utilidade neste sistema, contudo outros há que se afiguram completamente adequados.

A adopção da interface proposta anteriormente, tem também a vantagem, de não ser necessário o desenvolvimento de outro módulo para tornar a interface do sistema conforme a interface aceite pela plataforma de *middleware*.

Na Figura 5.6 é apresentado o conjunto de métodos, na forma de uma classe, a classe FrontEnd, que implementa o componente "Front-End" e fornece o acesso programático ao Agregador de Registos Bibliográficos.

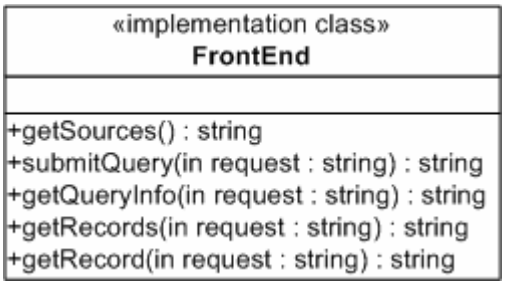



Figura 5.6 – Classe FrontEnd.

Por comparação, verifica-se então que a classe FrontEnd implementa um subconjunto dos métodos propostos na interface SpriInterface. Os restantes métodos, que não são aqui implementados, encontram-se fora do alcance funcional do sistema. Apenas para citar um exemplo, o método getIndex() não pode ser implementado aqui, porque a funcionalidade que lhe está associada não encontra paralelo em toda a funcionalidade oferecida pelos servidores Z39.50.

Em relação aos parâmetros dos métodos da classe FrontEnd, estes aceitam precisamente o mesmo tipo de dados, que os seus congéneres pertencentes à interface SpriInterface. Cabe, neste caso, à classe FrontEnd receber, validar e interpretar os documentos XML de pedido e gerar os documentos XML de resposta adequados.

#### 5.4.1.6 A Base de Dados

Foi explicado atrás que os componentes "Processador de Duplicados" e "Front-End" executam operações sobre uma base de dados, onde residem os conjuntos de registos, resultado das operações de pesquisa do sistema. Foi explicado também que os registos são convertidos para Dublin Core, sobre suporte XML.

Perante este cenário, foi escolhida uma base de dados nativa XML, por permitir a salvaguarda de documentos XML no seu formato original, assim como a pesquisa e manipulação directa dos elementos constituintes desses documentos.

Apesar da existência de múltiplas ofertas, comerciais e não comerciais, deste tipo de bases de dados, foi escolhida a base de dados Xindice (Apache, 2007) da fundação Apache. Este software, apesar de conter algumas lacunas em diversas funcionalidades, serve o propósito do sistema e possui duas vantagens primordiais: é gratuito, não implicando custos, e é desenvolvido por um grupo de elevado renome, no domínio do *open source*, o que facilita o suporte da aplicação nos casos de surgimento de problemas.

##### *A Interface*

Na interacção com a base de dados, pretendeu-se manter o maior grau possível de independência face a esta. A vantagem desta aproximação é óbvia: a não dependência de uma base de dados particular. Foi com este objectivo em mente que se decidiu pela utilização da interface XML:DB (XmlDB, 2003), com a qual a Xindice é compatível.

### *A Pesquisa*

Sobre as possibilidades de pesquisa desta base de dados. O componente "Processador de Duplicados" necessita de proceder a comparações entre registos. Para isso, necessita de capacidade de pesquisa sobre os registos que se encontram na base de dados. A base de dados Xindice permite a utilização da linguagem XPath (W3C, 1999d) para selecção de nós num documento XML. Apesar de não se tratar de uma linguagem de pesquisa na verdadeira acepção da palavra, oferece a funcionalidade suficiente para o tipo de pesquisa em causa.

As duas principais expressões de pesquisa utilizadas pelo componente "Processador de Duplicados" são:

- "/ua:record/dc:identifier[contains(.,id)]";
- e "/ua:record[dc:title='title']".

A primeira expressão é utilizada no primeiro método de identificação de registos duplicados e permite identificar todos os registos que contenham um determinado identificador (ISBN, ISSN ou outro) no elemento dc:identifier.

A segunda expressão é utilizada no segundo método, mais moroso por implicar a comparação de múltiplos elementos, e permite identificar todos os registos em que o título (dc:title) é igual a uma dada *string*.

#### 5.4.1.7 A Parametrização

Até ao momento foram apresentadas as escolhas e soluções técnicas adoptadas para a fase de desenvolvimento do módulo SPD. Na fase de execução, o módulo necessita de informação de parametrização, por forma a colocar em prática algumas das opções tomadas na fase mencionada anteriormente.

A informação de parametrização consiste em informação que poderá variar no decorrer do tempo de vida do sistema implementado. Por isso, é geralmente localizada fora do próprio sistema e acessível a um agente exterior, como um administrador de sistema por exemplo, que poderá proceder a modificações e actualizações dessa informação.

A informação de parametrização do módulo SPD reside num documento XML, perfeitamente inteligível pelo ser humano e editável através de um simples editor de texto. Na Figura 5.7 encontra-se parcialmente ilustrado este documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
 <logfile>conf\\z3950\\z3950.log</logfile>
 <querytimeout>60</querytimeout> <!-- seconds -->
 <querytimetolive>30</querytimetolive> <!-- minutes -->
 <sources>
  <source uri="urn:opacs:brunel">
   <name>Universidade de Brunel</name>
   <location>library.brunel.ac.uk:2200</location>
   <user>anonymous</user>
   <password>anonymous@email.org</password>
   <databases>
    <database>unicorn</database>
   </databases>
   <syntaxrec>usmarc</syntaxrec>
   <querymaxrec>50</querymaxrec>
   <active>1</active>
  </source>
  <source uri="urn:opacs:aberdeen">
   <name>Universidade de Aberdeen</name>
   <location>library.abdn.ac.uk:210</location>
   <databases>
    <database>dynix</database>
   </databases>
   <syntaxrec>usmarc</syntaxrec>
   <querymaxrec>50</querymaxrec>
   <active>1</active>
  </source>
  ...
 </sources>
 <normalization>
  <conversion from="marc21">
   conf\\z3950\\res\\MARC21slim2DC.xsl
  </conversion>
  <conversion from="unimarc">
   conf\\z3950\\res\\Unimarc2DC.xsl
  </conversion>
 </normalization>
</config>
```

Figura 5.7 – Documento de parametrização do módulo SPD.

Atentando na Figura 5.7, a informação de parametrização encontra-se organizada em três partes principais: a primeira, de índole geral; a segunda, orientada às fontes de informação ou servidores; e a terceira, orientada ao processo de normalização.

Na primeira parte encontram-se têes elementos com o seguinte significado:

- logfile – nome e localização do ficheiro onde ficam registados os eventos do módulo por forma a permitir uma depuração do seu funcionamento;
- querytimeout – tempo máximo, em segundos, para a execução de uma pesquisa;

- querytimetolive – tempo máximo, em minutos, para a permanência temporária dos resultados de uma dada pesquisa na base de dados.

Na segunda parte, dentro do elemento “sources”, podem encontrar-se vários elementos “source”, que possuem uma composição inspirada no modelo de dados da plataforma de *middleware*, tendo sido adicionados mais alguns elementos devido à especificidade das fontes a aceder. O elemento “source” é assim constituído por:

- uri – atributo do elemento “source” que identifica univocamente o servidor, no contexto do módulo;
- name – descrição do servidor;
- location – endereço Internet do servidor;
- user – nome de utilizador para aceder ao servidor (opcional);
- password – palavra chave do utilizador referido (opcional);
- databases – pode conter uma ou várias bases de dados pertencentes ao servidor, nas quais se pretende executar a pesquisa;
- syntaxrec – formato de registos pretendido para as respostas do servidor, escolhido a partir do conjunto de formatos disponíveis pelo servidor;
- querymaxrec – número máximo de registos a recolher do servidor. Este número deve ser afinado pelo administrador do sistema, em face do número total de servidores a consultar pelo módulo. Não se pode pretender recolher todos os registos encontrados para cada servidor, pois rapidamente esse número chegaria à casa dos milhares, tornando incomportável o seu processamento e a sua utilidade bastante duvidosa. O valor deste parâmetro tem precedência sobre o parâmetro referido em ua:numrecs, no pedido de pesquisa, caso este último seja superior ao anterior;
- active – indica ao módulo se pode ou não dirigir pesquisas a este servidor (1 – sim; 0 – não). O administrador do sistema pode, por exemplo, descobrir que um determinado servidor se encontra temporariamente fora de serviço e nesse caso coloca este parâmetro a zero, evitando a constante tentativa do seu contacto.

É ainda de referir, em relação a esta segunda parte dos elementos, que esta informação poderia constituir um directório de serviços idêntico ao preconizado na arquitectura da plataforma de *middleware*. Apenas teria um senão: a sua actualização dependeria sempre de um administrador local, uma vez que os servidores Z39.50 não

oferecem, no seu leque de funcionalidades, a possibilidade de se registarem em qualquer directório.

A terceira, e última, parte é iniciada com o elemento “normalization”, podendo conter múltiplos elementos “conversion”. Este último elemento contem informação acerca do XSLT necessário para a conversão a partir de um determinado formato. A sua constituição é dada por:

- from – atributo do elemento “conversion” indicando o formato a partir do qual se pretende efectuar a conversão;
- conteúdo do elemento – nome e localização do documento XSLT apropriado à conversão do formato indicado para Dublin Core.

### 5.4.2 O Módulo WS

Como já foi descrito, a arquitectura do Agregador sugere que ambos os módulos, o SPD e o WS, sejam implementados sobre a plataforma JAVA.

#### 5.4.2.1 O *Web Service*

A implementação do *web service* para este sistema foi amplamente facilitada por dois factores:

- a sua interface encontrava-se inteiramente definida;
- e a implementação real dos métodos é efectuada fora do *web service*.

A interface oferecida por este *web service* é a que foi definida no capítulo anterior como a interface funcional comum para qualquer serviço a integrar na plataforma de *middleware*; e a implementação efectiva do serviço é a que reside no módulo SPD. De facto, o *web service*, neste caso, não é mais que um simples *wrapper* do módulo SPD.

A implementação do *web service* resume-se assim a instanciar a interface mencionada acima e para os métodos que já têm implementação no módulo SPD, basta efectuar a ligação aos mesmos. Para os métodos que não têm implementação, o *web service* limita-se a gerar uma mensagem de erro. O único método, que não tem implementação no módulo SPD e que deve ser implementado no próprio *web service* é o método hello(). Este método, como foi explicado na sua apresentação, é unicamente destinado a averiguar da existência ou operacionalidade do *web service*.

#### 5.4.2.2 Suporte ao *Web Service*

No caso específico do módulo WS, não foi a sua implementação que colocou desafios, mas sim a escolha do seu suporte. A implementação de um *web service*, em qualquer que seja a plataforma, carece de uma aplicação ou ambiente de suporte, uma vez que esse serviço utiliza capacidades de comunicação na rede. Geralmente, a aplicação que fornece esse serviço é um servidor web, com capacidades para mais que simplesmente servir páginas HTML.

Apesar das diferentes alternativas, optou-se mais uma vez por uma solução *open source*. Foram escolhidos, as aplicações pertencentes à Apache Foundation:

- o AXIS (Apache, 2006), que consiste numa implementação do SOAP;
- e o Tomcat (Apache, 2009a), para servir de plataforma de execução ao *web service*.

A razão da escolha destas aplicações, é basicamente a mesma que levou à escolha da base de dados Xindice. São aplicações pertencentes a uma organização do domínio *Open Source* com créditos firmados internacionalmente; e não necessitam de bibliotecas de classes de outras organizações, que não as que suportam a própria JVM.

## 5.5 A Interface com o Utilizador

Como referido na secção sobre requisitos, foi colocada à partida a pretensão de desenvolver uma interface de utilizador para o Agregador, inteiramente baseada na web. São apresentados nesta secção: a arquitectura; as opções tecnológicas; e a ilustração dessa interface.

### 5.5.1 Arquitectura e Soluções Técnicas

Para a implementação de uma interface de utilizador baseada na web foi concebida uma arquitectura e seleccionadas algumas soluções técnicas que, para além de permitir o acesso imediato do Agregador aos utilizadores, permitiu também testar e validar parcialmente o modelo da plataforma de *middleware*, assim como um dos pressupostos de todo o trabalho: a integração de sistemas distribuídos heterogéneos.

Na Figura 5.8 encontra-se representada a arquitectura que norteou o desenvolvimento desta interface de utilizador.

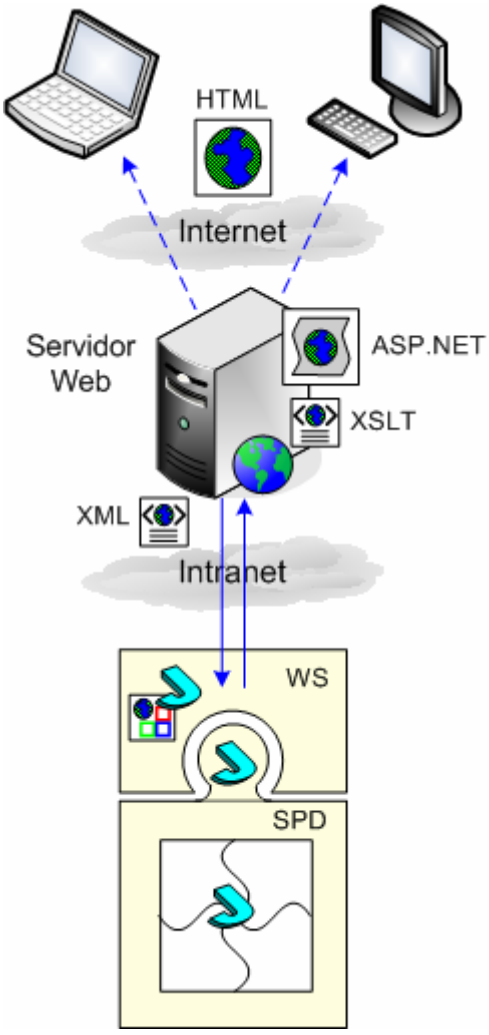



Figura 5.8 – Arquitectura da interface de utilizador.

Como descrito anteriormente, e se denota pela Figura 5.8, os módulos que implementam o Agregador foram desenvolvidos sobre a plataforma JAVA. Para implementação da interface de utilizador foram seleccionados um servidor web, que executa sobre a plataforma Windows, e a plataforma .NET (Microsoft, 2009a) para implementação de aplicações que executam no *back-end* do servidor.

Na riqueza da combinação destas diferentes tecnologias e plataformas encontra-se a possibilidade confirmada da integração de sistemas heterogéneos perseguida por este trabalho.

As aplicações *back-end* do servidor web, podem ser vistas como um SPRI de uma determinada camada que utiliza os modelos funcionais e de dados, apresentados no capítulo anterior, para aceder a um SPRI da camada imediatamente abaixo. Esta

interpretação evidencia assim a concretização parcial do modelo concebido anteriormente para a plataforma de *middleware*.

### 5.5.2 Desenvolvimento

Para além da utilização das plataformas e tecnologias mencionadas atrás, foram desenvolvidas algumas aplicações *back-end*, para o servidor web, responsáveis:

- pela recepção e interpretação dos pedidos dos utilizadores;
- o envio dos pedidos ao Agregador;
- a recepção dos resultados da parte do Agregador;
- a geração da representação dos resultados;
- e o envio das representações, no formato HTML, aos utilizadores.

```
<?xml version='1.0'?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:ua="http://www.ua.pt/spri/results">
 <xsl:output method="html"/>
 <xsl:template match="/ua:results">
  <script language="jscript">
   function setOperation(str)
   {
    document.ResForm.hdOp.value = str;
    document.ResForm.submit();
   }
  </script>
  <table border="0" class="datainfo" width="600"
         cellspacing="1" cellpadding="0" align="center">
   <xsl:apply-templates select="//ua:info" mode="header"/>
  </table>
  <table border="0" width="600" cellspacing="1" cellpadding="0" align="center">
   <tr class="th">
    <th width="3%" align="center">&amp;#35;</th>
    <th width="45%" align="center">Title</th>
    <th width="20%" align="center">Author</th>
    <th width="7%" align="center">Date</th>
    <th width="22%" align="center" class="lastcol">Location</th>
   </tr>
   <xsl:apply-templates select="//ua:record"/>
  </table>
 ...
 </xsl:template>
</xsl:stylesheet>
```

Figura 5.9 – XSLT responsável pela geração da representação das respostas.

As aplicações *back-end* são basicamente de dois tipos:

- um componente, o Z3950WSClient, desenvolvido como uma *Class Library*, que permite a sua reutilização em outras aplicações e responsável pela interacção com o Agregador, via *web services*. Este componente é responsável pela criação dos documentos XML de pedido que são enviados ao Agregador e pela recepção dos documentos de resposta;
- várias ASPX, que utilizam a tecnologia ASP.NET (Microsoft, 2009b), responsáveis por gerar as páginas HTML que interagem com os utilizadores.

Os documentos XML de pedido gerados pelo componente Z3950WSClient são documentos completamente conformes aos modelos apresentados no capítulo anterior.

A geração das páginas HTML, que constituem a representação das respostas enviadas pelo Agregador, efectuada pelas ASPX é concretizada através de documentos XSLT, que transformam automaticamente os documentos XML recebidos em páginas HTML. Uma amostra desses documentos XSLT encontra-se ilustrada na Figura 5.9.

### 5.5.3 A Interface Gráfica

De seguida é descrita a interface gráfica, propriamente dita. Esta interface é composta por três tipos base de páginas: a página de entrada; a página de resultados e a página de registo.

Todas as páginas são encimadas por uma imagem de logótipo a qual possui a inscrição “Bibliotheca Universalis”. Não se pretende, neste trabalho, a apropriação do termo, mas sim apenas relembrar uma velha pretensão da humanidade em construir uma biblioteca que detivesse todo o conhecimento alguma vez criado pelo ser humano. Não é certamente o caso deste Agregador de Registos Bibliográficos, mas fica desta forma traduzida a esperança de um dia tal vir a ser possível, sendo para isso necessárias todas as contribuições, por muito pequenas que sejam.

#### 5.5.3.1 A Página de Entrada

A página de entrada do Agregador apresenta de imediato o formulário que permite ao utilizador efectuar os seus pedidos de pesquisa e encontra-se ilustrada na Figura 5.10.

Na primeira parte, o formulário permite a introdução da pesquisa propriamente dita, possibilitando introduzir os termos de pesquisa, seleccionar os índices ou campos a que pertencem os termos e seleccionar os operadores de conjunção que pretende, no caso de pesquisas compostas. Os campos pelos quais é possível pesquisar são: título, autor,

assunto, editor, data, ISBN, ISSN ou todos. Os operadores permitidos são: e, ou, e não. Estas possibilidades de pesquisa são muito parecidas com as da maioria dos OPACs que se podem encontrar. Não são por isso qualquer novidade para um utilizador habituado a pesquisar na sua biblioteca favorita.

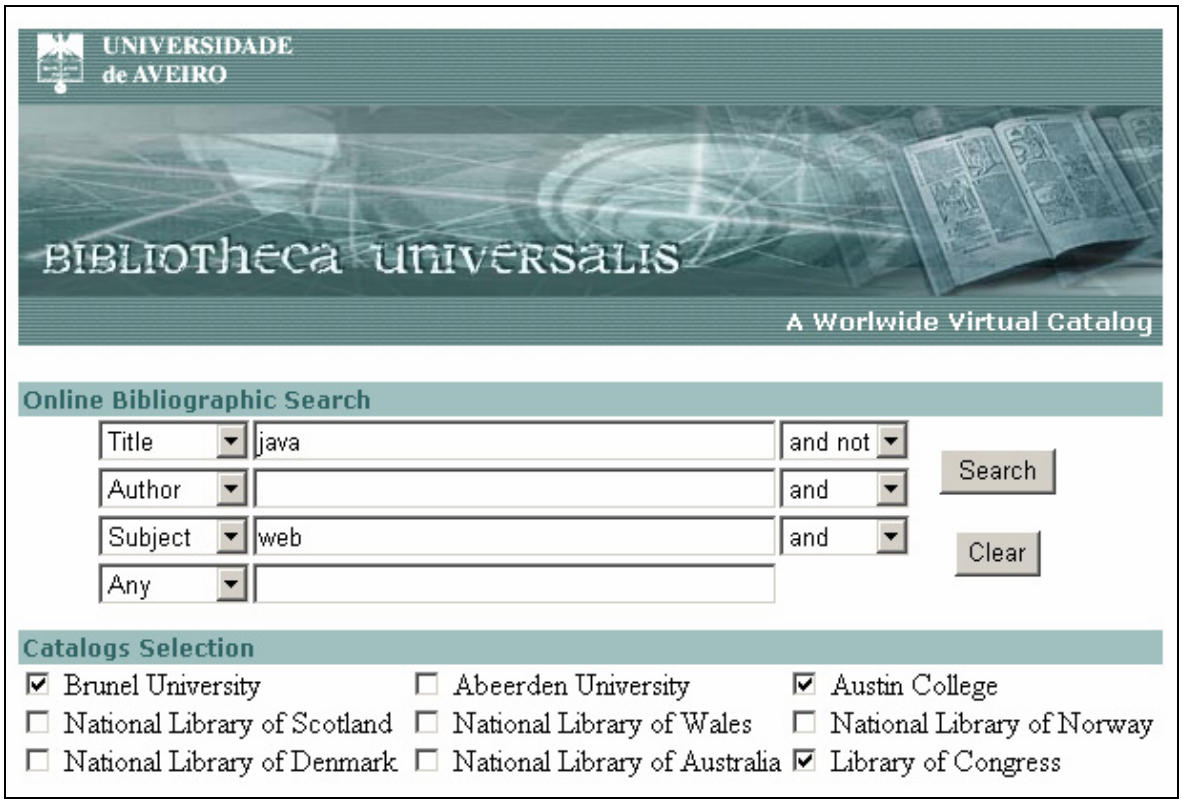



Figura 5.10 Página de entrada do Agregador de Registos Bibliográficos.

Na segunda parte, o utilizador encontra-se presente a um conjunto de bibliotecas que pode seleccionar como destino da sua pesquisa. Caso seleccione alguma ou algumas, a pesquisa será dirigida apenas às seleccionadas, caso não seleccione qualquer biblioteca, o sistema fará a pesquisa em todas elas.

No caso particular da Figura 5.10, o utilizador pretende efectuar uma pesquisa composta em que o termo “java” deverá aparecer no campo “título” e o termo “web” não deverá aparecer no campo “assunto”. São ainda seleccionadas as bibliotecas “Brunel University”, “Austin College” e “Library of Congress” como destino da pesquisa.

#### 5.5.3.2 A Página de Resultados

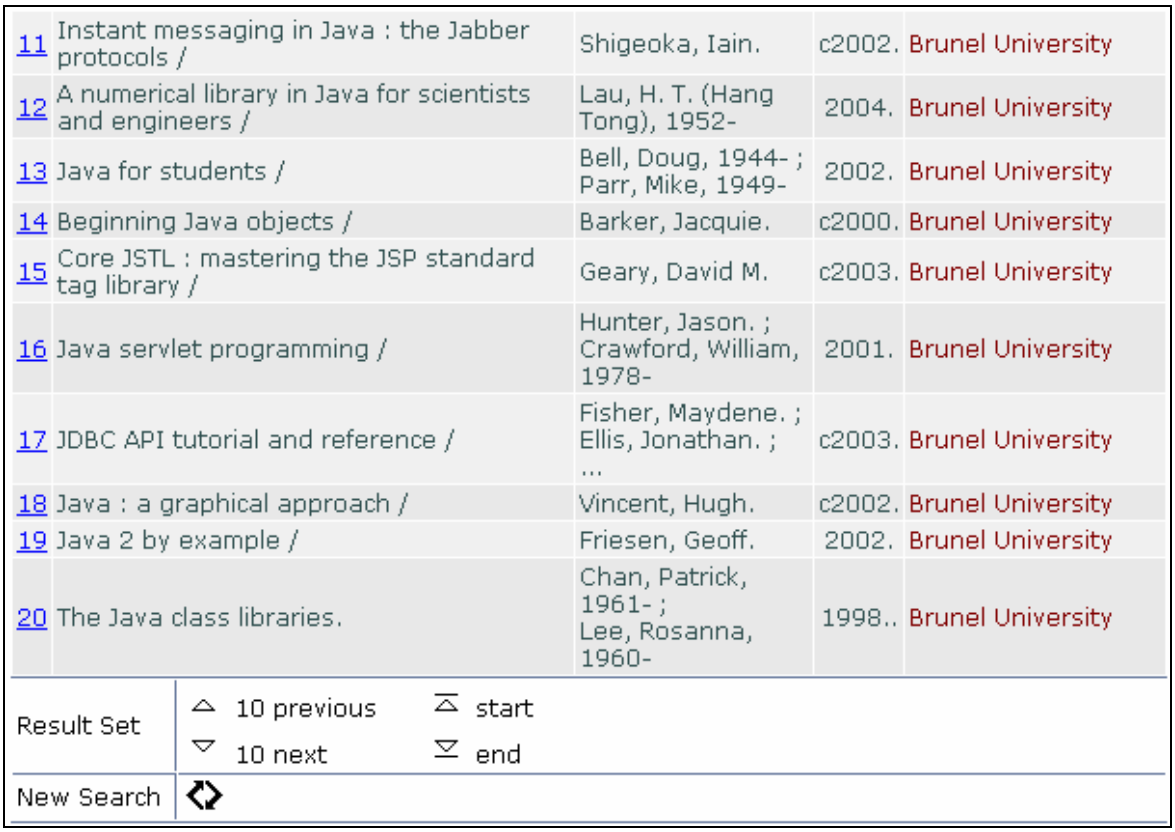
O resultado da pesquisa anterior pode ser visualizado na Figura 5.11 e Figura 5.12.


Figura 5.11 – Parte superior da página de resultados do Agregador.

Na parte superior da página de resultados são afixados alguns dados sobre a pesquisa:

- a pesquisa efectuada;
- o número total de registos encontrados nas bibliotecas que satisfazem a pesquisa;
- os números dos registos constantes da página de resultados actual;
- o número total de registos recolhidos das bibliotecas;

Na parte central da página aparecem então os sumários dos registos, constituídos pelo título, autores, data e biblioteca ou bibliotecas onde se encontram.



| | | | | |
|---|---|---|---|---|
| 11 | Instant messaging in Java : the Jabber protocols / | Shigeoka, Iain. | c2002. | Brunel University |
| 12 | A numerical library in Java for scientists and engineers / | Lau, H. T. (Hang Tong), 1952- | 2004. | Brunel University |
| 13 | Java for students / | Bell, Doug, 1944- ; Parr, Mike, 1949- | 2002. | Brunel University |
| 14 | Beginning Java objects / | Barker, Jacquie. | c2000. | Brunel University |
| 15 | Core JSTL : mastering the JSP standard tag library / | Geary, David M. | c2003. | Brunel University |
| 16 | Java servlet programming / | Hunter, Jason. ; Crawford, William, 1978- | 2001. | Brunel University |
| 17 | JDBC API tutorial and reference / | Fisher, Maydene. ; Ellis, Jonathan. ; ... | c2003. | Brunel University |
| 18 | Java : a graphical approach / | Vincent, Hugh. | c2002. | Brunel University |
| 19 | Java 2 by example / | Friesen, Geoff. | 2002. | Brunel University |
| 20 | The Java class libraries. | Chan, Patrick, 1961- ; Lee, Rosanna, 1960- | 1998.. | Brunel University |



Figura 5.12 – Parte inferior da página de resultados do Agregador.

Na parte inferior da página, visível na Figura 5.12, encontram-se presentes diversos "botões" que permitem a navegação no conjunto de resultados obtido:

- os dez anteriores;
- os dez seguintes;
- a primeira página;
- a última página;
- recomeçar uma nova pesquisa.

#### 5.5.3.3 A Página de Registo

Os registos apresentados na página de resultados são apenas sumários dos registos, isto é, uma selecção de alguns dos campos mais relevantes dos registos. Para a visualização dos registos na sua totalidade, o utilizador apenas tem de seleccionar o registo que pretende visualizar, pressionando com o cursor do rato sobre o número do registo em causa, e será apresentado o registo na sua totalidade na página de registo.

Na Figura 5.13 apresenta-se na página de registo o registo nº 13, do conjunto de registos visualizado na página de resultados.

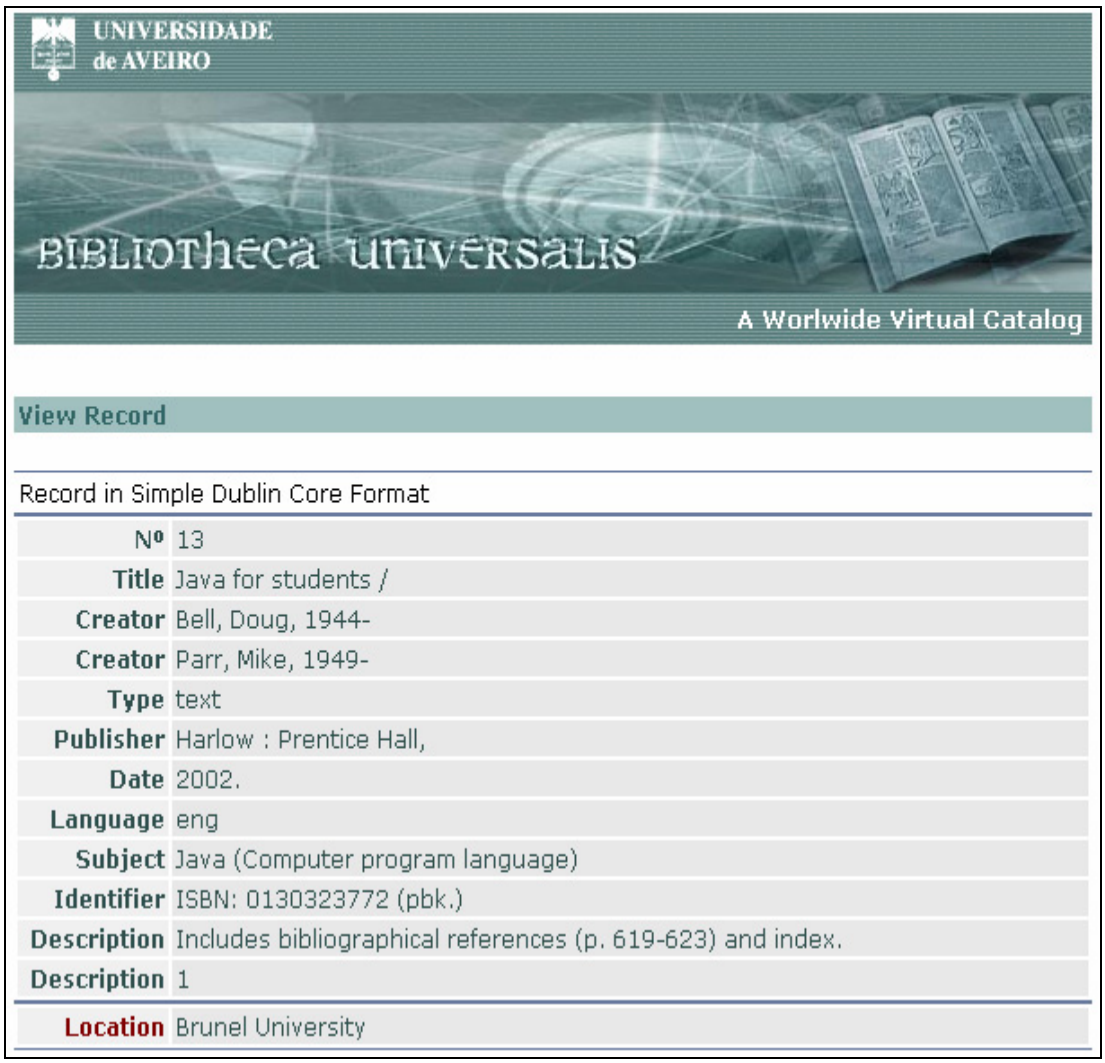



Figura 5.13 – Página de Registo do Agregador.

Nesta página é visível a separação da informação de localização do registo, pertencente à "Brunel University", da restante informação. Este aspecto, apesar de constituir apenas uma questão de apresentação da informação, é considerada de primordial importância, uma vez que a informação de localização do registo é informação exterior ao registo. Esta é-lhe adicionada à posteriori quando chega ao Agregador e tem apenas por missão contextualizar a sua origem.

## 5.6 Revisão

O presente capítulo descreve um sistema intitulado Agregador de Registos Bibliográficos, como sistema integrante da plataforma de *middleware* descrita no capítulo anterior. Neste contexto, este sistema é apenas um entre muitos que contribui para o enriquecimento da informação final a disponibilizar pela plataforma.

A descrição do sistema inicia-se por uma enunciação de alguns requisitos tidos como preponderantes para a sua concepção e implementação. É seguidamente apresentada a sua arquitectura como modelo para o seu desenvolvimento. São também descritas as opções e soluções técnicas desse desenvolvimento. Por fim, é apresentada a interface de utilizador desenvolvida particularmente para este sistema, como meio de disponibilizar de imediato a sua funcionalidade ao utilizador final.

A concepção deste sistema foi baseada, em larga medida, nos conceitos e modelos apresentados para a plataforma de *middleware* no capítulo anterior. Podendo-se, com as devidas salvaguardas, relativamente à dimensão e às tecnologias usadas, assumir também este sistema como uma plataforma de *middleware* orientada para fontes de informação específicas e idênticas. De facto, este sistema faz a um nível particular o que se pretende que a plataforma de *middleware* faça num nível mais abrangente. Podendo-se, por isso, pensar em cada SPRI do modelo de abstracção como uma plataforma de *middleware* em dimensões mais reduzidas.

É esta perspectiva que permite tomar este sistema por um demonstrador da plataforma de *middleware*, salvaguardando as dimensões, e o conhecimento do seu comportamento funcional pode, até certo ponto, fornecer indicadores de previsão relativamente ao comportamento da plataforma.

O desenvolvimento do Agregador, em si mesmo, revelou que tal sistema é exequível, apesar dos desafios de engenharia colocados à partida, e que a utilização das tecnologias e padrões abertos actuais contribuem enormemente para interoperabilidade dos sistemas heterogéneos utilizados, dando como exemplo, os *web services*.

A utilização do Dublin Core, como formato de metadados para a normalização dos formatos dos registos, foi também considerada um êxito, visto não só se ter revelado possível a sua aplicação, como a sua simplicidade ter acabado por trazer ao sistema vantagens de ordem processual e vantagens ao utilizador final, através de uma melhor compreensão da informação recolhida.

# Capítulo 6

# Repositórios de Informação

Os repositórios de informação são os sistemas que detêm de facto a informação e que procedem ao seu armazenamento com vista à sua posterior utilização e preservação.

Apesar da enorme oferta de sistemas presentes no mercado para a implementação de repositórios de informação, nomeadamente as inúmeras bases de dados provenientes de diferentes fornecedores e que utilizam diferentes abordagens de armazenamento, considerou-se neste trabalho estender o estudo a estes sistemas com vista à apresentação de propostas que possibilitassem: o armazenamento, pesquisa e recolha mais eficazes da informação; e o seu acesso distribuído com menores problemas de interoperabilidade.

Na arquitectura da plataforma de *middleware*, descrita no capítulo 4, os SPRIs da terceira camada são elementos funcionais que consistem basicamente em repositórios de informação, oferecendo uma interface de pesquisa e uma interface de comunicação remota para permitir a distribuição dos próprios repositórios.

Neste capítulo são primeiramente identificados os diferentes tipos de informação residentes nos repositórios de informação; é depois abordada a problemática da utilização de diferentes soluções para o armazenamento de dados XML; é descrito o *middleware* para acesso remoto, implementado para dois tipos diferentes de repositórios de informação, um baseado numa base de dados XML nativa e outro baseado num

sistema de ficheiros e indexador; e por fim é descrito o *middleware* implementado para melhorar a indexação do formato XML no repositório baseado no sistema de ficheiros.

## 6.1 A Informação

Procedendo a uma análise do tipo de informação que geralmente se pretende armazenar num repositório de informação de suporte a uma biblioteca digital, identificam-se os seguintes tipos de informação:

- textual, não estruturada (ficheiros de texto simples);
- textual estruturada (ficheiros XML contendo dados e/ou metadados);
- imagens;
- sons;
- vídeos.

A informação textual não estruturada pode ser, ou não, uma cópia textual de informação que se encontra em outros formatos, como: imagens, sons ou descrições de vídeo. O seu principal objectivo é proporcionar a pesquisa em texto livre sobre essa informação.

A informação textual estruturada em XML, orientada ao documento ou aos dados, pode constituir informação, em si mesma, tal como a informação não estruturada, ou pode constituir informação descritiva de outra informação, sendo considerada de metadados. Em qualquer dos casos, a informação nesta forma tem por principal objectivo a pesquisa sobre índices específicos, como: título, autor, assunto, data, descrição, etc.

Os restantes formatos informativos (imagens, sons e vídeos) possuem carácter informativo ao utilizador final, mas não constituem motivo de pesquisa automática, excepção feita aos nomes dos ficheiros que os comportam. A pesquisa automática da informação contida nestes formatos é efectuada sobre eventuais textos, estruturados ou não, que a descrevam.

## 6.2 Bases de Dados

A utilização de bases de dados é sempre tentadora, quando se trata do armazenamento de informação sobre a qual irão pesar posteriormente operações de modificação, actualização e, sobretudo, pesquisa. Esta última operação, em particular, é sobretudo dirigida aos tipos de informação textual, estruturada ou não.

As actuais bases de dados relacionais permitem actualmente um vasto leque de operações sobre qualquer um dos tipos de informação mencionados acima. Contudo, sem ter em conta os artifícios mais ou menos elaborados que algumas bases de dados põem em prática, o modelo relacional clássico deixa antever diversos problemas que surgirão inevitavelmente na manipulação de informação no modelo XML. Uma estrutura XML pode apresentar uma tal complexidade e profundidade, que a sua real partição num modelo com tabelas relacionais pode ser geradora de uma muito maior complexidade e decorrente daí, uma muito maior dificuldade no seu tratamento.

As bases de dados XML nativas, surgiram precisamente para tentar evitar este problema, pois não procedem a qualquer conversão do modelo XML para outro modelo e armazenam estes documentos como entidades individuais, sem recorrer à partição da sua estrutura. Várias das actuais bases de dados deste tipo permitem também a salvaguarda dos outros tipos de informação, mencionados acima.

### 6.2.1 A Interface XML:DB

A interface XML:DB (XmlDB, 2003), como descrita anteriormente, consiste num meio de acesso que se pretende afirmar como um padrão para o acesso de bases de dados XML nativas. Em essência, a interface XML:DB não é mais que uma especificação, a sua implementação está a cargo de cada base de dados que a oferece.

Algumas das bases de dados XML nativas oferecem esta interface de acesso, como uma API, e oferecem ainda outros tipos de acesso como a Xindice, por exemplo, que oferece também um acesso XML-RPC (XMLRPC, 2009), ou a eXist (eXist, 2009), que oferece ainda uma maior gama de acessos.

Apesar das vantagens que podem advir da oferta de múltiplos tipos de acesso a estas bases de dados, julga-se que a proliferação indiscriminada de meios de acesso acaba por contribuir para uma maior confusão, quando é chegado o momento de fazer uma escolha.

Com este pensamento em mente e julgando-se a interface XML:DB uma interface equilibrada e adaptável, considerou-se desenvolver um *web service* para esta interface.

### 6.2.2 XML:DB *Web Service*

A interface XML:DB, em si, já prevê o acesso remoto e, por isso, a distribuição. Nesta perspectiva, poderá questionar-se a razão para o desenvolvimento de um *web service* para esta interface.

De facto, um *web service* é muito mais que um simples método para distribuir aplicações. É um si um meio padronizado de acesso, que utiliza tecnologias abertas e padrão, permite o seu registo extensivo e permite a uma aplicação utilizar com relativa facilidade uma nova interface, desconhecida até ao momento.

Este *web service* foi desenvolvido sobre a plataforma JAVA e a representação do seu modelo WSDL encontra-se na Figura 6.1.

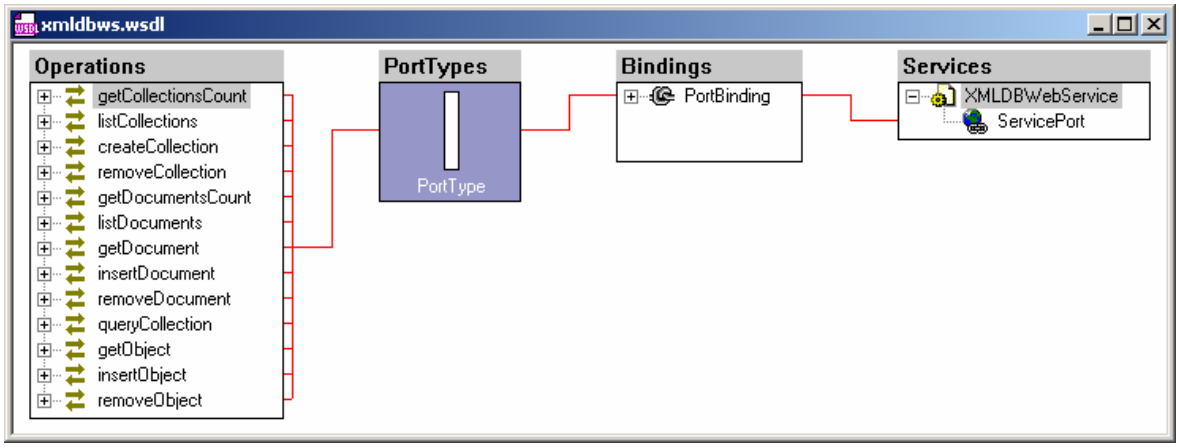



Figura 6.1 – Representação do modelo WSDL do XML:DB *web service*.

O *web service* desenvolvido para a interface XML:DB não pretende mapear toda a funcionalidade da mesma, mas sim oferecer uma interface de alto nível, que permita a execução das principais e mais requisitadas operações. Na Figura 6.2 é possível ver, em mais detalhe, as operações e as mensagens que as compõem.

Existem três grupos principais de operações, que se dividem em função do objecto sobre o qual actuam:

- o grupo de operações sobre colecções;
- o grupo de operações sobre documentos XML;
- o grupo de operações sobre objectos binários.

### 6.2.2.1 Grupo de Operações sobre colecções

A primeira operação, a operação “getCollectionCount”, tem por objectivo devolver o número de colecções existentes dentro de uma dada colecção. O parâmetro de entrada é do tipo *string* e identifica a colecção “mãe”. O parâmetro de saída é do tipo inteiro.

A segunda operação, a operação “listCollections”, devolve uma lista de nomes de colecções existentes dentro da uma dada colecção. No parâmetro de entrada é

designada a colecção "mãe". O parâmetro de saída, do tipo *string*, consiste num documento XML muito simples, contendo a listagem requerida. Na Figura 6.3 encontra-se ilustrado um exemplo de um documento desse tipo.

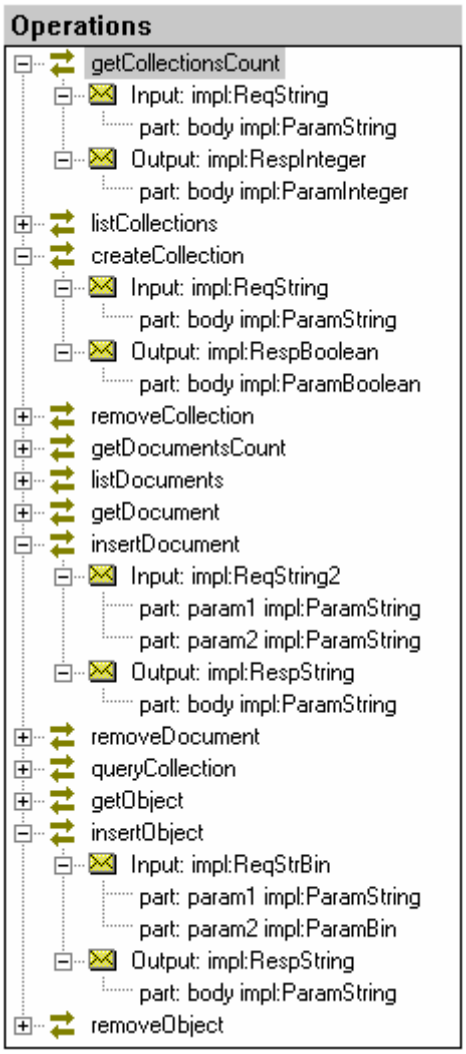



Figura 6.2 – Diagrama das operações do *web service*.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<results>
  <item>collection01</item>
  <item>collection02</item>
  <item>collection03</item>
  <item>collection04</item>
</results>
```

Figura 6.3 – Exemplo de um documento XML contendo uma listagem de colecções.

A terceira operação, a operação “createCollection”, permite a criação de uma colecção dentro de uma dada colecção. No parâmetro de entrada é dado o nome da colecção a criar e o parâmetro de saída, do tipo *boolean*, indica se a operação executou correctamente.

A quarta e última operação deste grupo, a operação “removeCollection”, permite remover uma dada colecção, assim como todo o seu conteúdo. O parâmetro de entrada fornece o nome da colecção a remover e o parâmetro de saída indica o sucesso ou não da execução da operação.

#### 6.2.2.2 Grupo de Operações sobre documentos

A primeira operação deste grupo, a operação “getDocumentsCount”, possui uma funcionalidade idêntica à da operação “getCollectionsCount”, mas neste caso devolve o número de documentos existentes numa dada colecção.

A operação “listDocuments” devolve a lista dos nomes dos documentos existentes numa dada colecção. A forma apresentada por essa lista é idêntica à do exemplo da Figura 6.3.

A operação “getDocument” permite a recolha do conteúdo de um determinado documento, dado o seu nome e colecção a que pertence. Neste caso, é dado o caminho absoluto para o documento, indicando toda a hierarquia de colecções até chegar ao documento, tal como de um sistema de ficheiros se tratasse.

A operação “insertDocument” permite a inserção de um novo documento numa dada colecção. Neste caso são utilizados dois parâmetros de entrada: um, para indicar o caminho absoluto do documento; outro, para transportar o conteúdo do mesmo. Caso o documento já exista, será reescrito.

A operação “removeDocument” permite remover um determinado documento de uma determinada colecção.

A operação “queryCollection” permite submeter uma pesquisa à base de dados. Dependendo das capacidades da base de dados, a linguagem de pesquisa poderá ser XQuery (W3C, 2007b) ou simplesmente XPath. A lista de resultados devolvida toma a forma do exemplo já mencionado acima para a operação “listDocuments”.

#### 6.2.2.3 Grupo de Operações sobre Objectos Binários

Este grupo de operações consiste em apenas três operações: “getObject”, “insertObject” e “removeObject”. Estas operações possuem funcionalidades similares

àquelas que manipulam documentos, com excepção para o parâmetro que transporta o conteúdo que é do tipo binário.

Neste grupo não há lugar a operações de listagem, pois o acesso a estes objectos é sempre efectuado a partir da sua referência nos documentos XML.

## 6.3 Sistemas de Ficheiros

Os sistemas de ficheiros são um dos meios mais antigos, tradicionais e simples de guardar informação em sistemas informáticos. A sua fiabilidade, como meio de armazenamento, e a sua eficiência, como meio para a pesquisa de informação, têm sido consideradas baixas, comparativamente às bases de dados. Contudo, utilizando algum grau de redundância e aplicações que permitam uma eficiente extracção da informação, os sistemas de ficheiros poderão revelar-se como um meio bastante poderoso para a salvaguarda de informação, principalmente em cenários de arquivo.

Dentro desta perspectiva, decidiu-se fazer uso deste sistema de armazenamento de informação e averiguar da sua capacidade para responder aos desafios colocados em diversos cenários práticos.

### 6.3.1 O Index Server da Microsoft

A aplicação Index Server da Microsoft é uma das aplicações exemplo, que podem ser utilizadas na extracção de informação localizada em sistemas de ficheiros. Para além desta, pode-se ainda referir a Apache Lucene (Apache, 2009b) como uma das mais utilizadas actualmente.

A opção recaiu sobre a Index Server por se tratar de um sistema que é instalado por defeito pelo sistema operativo Windows; ao tratar-se de uma aplicação Windows, esta iria dar a oportunidade de averiguar as possibilidades de interoperabilidade com outros sistemas; e de qualquer forma, ambas as aplicações não possuem capacidades nativas para o tratamento da estrutura XML.

Apesar das excelentes capacidades de indexação do Index Server e da boa qualidade dos resultados obtidos através das pesquisas possíveis através dele, este possui uma lacuna ao nível da disponibilização de informação: não permite a uma entidade exterior, pedir ou aceder a um índice de valores relativos a uma propriedade. Se por exemplo, se pretender ter acesso a uma lista de todos os autores presentes nos documentos indexados, não é possível. Isto, para todos os documentos reconhecidos pelo Index Server ou pelos seus filtros.

Os documentos XML, não se encontram entre os chamados documentos reconhecidos pelo Index Server. Podem, no limite, ser indexados como ficheiros de texto livre, o que implica: ignorar por completo toda a estrutura XML e a semântica associada aos seus elementos. Por esta razão, no presente trabalho, esta aplicação foi utilizada em conjunto com um filtro comercial, o QLXFilter (QuiLogic, 2009), que reconhece o formato XML.

Mais uma vez, apesar das excelentes capacidades do filtro QLXFilter para a extracção de informação de documentos XML, verificou-se a existência de uma lacuna no seu funcionamento, responsável pela inexistência de funcionalidades de elevada importância na aplicação Index Server. A lacuna detectada relaciona-se com elementos repetidos no mesmo nível hierárquico da estrutura XML. Apesar de conseguir detectar os seus diversos valores, não consegue transmitir essa informação de forma individualizada ao Index Server. Uma solução para tal, poderia ter sido a adopção da estratégia utilizada pelo filtro de HTML, que na presença de múltiplos elementos “meta”, com o mesmo nome e diferentes valores, transmite essa informação ao Index Server na forma de vectores de valores. Contudo, tal não foi considerado pelos fornecedores do filtro.

Considerando então as limitações, tanto do Index Server como do filtro QLXFilter, verificou-se as seguintes repercussões na funcionalidade oferecida por este par de aplicações:

- incapacidade para devolver valores múltiplos e individualizados respeitantes a uma determinada propriedade;
- incapacidade para devolver um índice correcto de valores, relativo também a uma propriedade específica.

Perante este cenário, foi julgado adequado o desenvolvimento de um componente, que fazendo uso de determinados artifícios consegue oferecer as funcionalidades que faltam ao Index Server, no caso específico da manipulação de documentos XML. Na prática este componente consiste num *wrapper* da aplicação Index Server, oferecendo funcionalidade adicional, tendo por isso sido chamado IndexsrvXWrapper.

### 6.3.2 O Componente IndexsrvXWrapper

O componente IndexsrvXWrapper apresenta uma funcionalidade similar ao Index Server, sendo o seu acesso efectuado nos mesmos moldes que o acesso ao Index Server. Na sua implementação foi feito o esforço para que a sua interface com o exterior seguisse o mais possível a interface original do Index Server. Desta forma, uma aplicação cliente do Index Server poderá utilizar o IndexsrvXWrapper da mesma forma, tanto para

recolher informação de documentos XML como de outros documentos suportados pelo Index Server e os seus filtros.

#### 6.3.2.1 Implementação do Componente

O componente IndexsrvXWrapper foi desenvolvido sobre a plataforma .Net na forma de uma Class Library. Não possui interface com o utilizador, mas apenas uma interface programática, tal como o próprio Index Server.

Na Figura 6.4 encontra-se representado o diagrama da classe que implementa este componente.

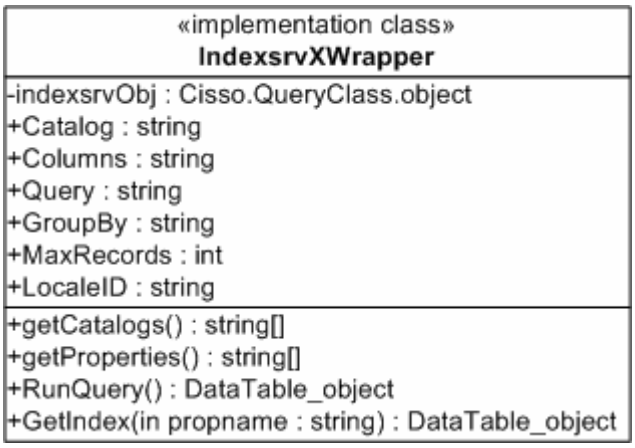



Figura 6.4 – Diagrama UML da classe IndexsrvXWrapper.

No diagrama e na zona dos atributos, os atributos visíveis não constituem toda a extensão de atributos presentes na classe mas sim os mais representativos. O primeiro atributo, o “indexsrvObj”, é um atributo privado e aparece aqui representado porque é o responsável pela ligação entre este componente e o Index Server. Os restantes atributos são públicos e aparecem representados para tentar transmitir o grau de fidelidade, à interface do Index Server, com que foi desenvolvido este componente.

Na zona dos métodos encontram-se apenas os métodos públicos e que podem ser invocados por aplicações exteriores. Os dois primeiros métodos podem ser usados para descobrir os catálogos e propriedades indexadas que se encontram presentes no Index Server.

O método “RunQuery” é responsável pela execução de uma pesquisa e devolve resultados na forma de uma DataTable que consiste num objecto pertencente às bibliotecas .Net do domínio das bases de dados. A *string* de pesquisa deve ser

previamente disposta no atributo “Query”, assim como os restantes atributos que também devem ser previamente configurados, antes da execução da pesquisa. Este procedimento é em tudo igual ao utilizado com o Index Server, mantendo assim a fidelidade à aplicação principal e evitando tempos desnecessários de aprendizagem.

O método “GetIndex” permite o pedido de um índice de valores relativamente a uma dada propriedade, que deve ser especificada como parâmetro. Este método representa uma funcionalidade nova, face à funcionalidade oferecida pelo Index Server.

#### 6.3.2.2 O Filtro QLXFilter e a Configuração do Componente

Para que seja possível ao componente fornecer funcionalidades que não se encontram presentes no Index Server, este tem de obedecer a uma rigorosa configuração em estrita conformidade com a configuração do filtro QLXFilter.

Em conjunto com o filtro, o fabricante fornece também uma ferramenta muito simples, com interface gráfica, que permite estabelecer previamente uma configuração para o funcionamento do mesmo. Nessa configuração é possível designar quais os elementos XML a considerar, para a extracção de informação, qual a expressão XQuery a utilizar nesse procedimento e por fim, sob que nome essa informação deverá aparecer no Index Server. A ferramenta guarda o resultado final da configuração num documento XML.

Para configuração do componente, também é utilizado um documento XML que, em função da configuração elaborada para o filtro, contém informação que contextualiza os elementos configurados para o mesmo. Na Figura 6.5 encontra-se ilustrado um exemplo desta configuração.

Analisando o exemplo referido, verifica-se que a configuração versa sobre a própria configuração das propriedades que se encontram disponíveis para acesso através do componente. Quando se pretende aceder programaticamente ao Index Server para efectuar pesquisas e recolher os eventuais resultados, são necessários alguns procedimentos prévios, como a definição, na aplicação que acede, de todas as propriedades a que vai tentar aceder. Para isso é necessário utilizar um conjunto de informação, como: o nome da propriedade, o seu tipo, o setguid a que pertence e o seu identificador.

Esta é a primeira razão para que o componente mantenha num ficheiro de configuração toda a informação respeitante às propriedades.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
 <catalogs>
  <catalog>
   <name>memafrica</name>
   <properties>
    <property>
     <name>title</name>
     <based>
      <indxsrvprop>
       <fname>marc200</fname>
       <datatype>(DBTYPE_WSTR|DBTYPE_BYREF)</datatype>
       <setguid>45807EF7-5D36-48CF-BCFE-596E15399DA7</setguid>
       <propid>marc200</propid>
       <customtype>value</customtype>
       <purpose>search retrieve</purpose>
      </indxsrvprop>
     </based>
    </property>
    <property>
     <name>subject</name>
     <based>
      <indxsrvprop>
       <fname>marc606txt</fname>
       <datatype/>
       <setguid>45807EF7-5D36-48CF-BCFE-596E15399DA7</setguid>
       <propid>marc606txt</propid>
       <customtype>text</customtype>
       <purpose>search</purpose>
      </indxsrvprop>
      <indxsrvprop>
       <fname>marc606xml</fname>
       <datatype>(DBTYPE_WSTR|DBTYPE_BYREF)</datatype>
       <setguid>45807EF7-5D36-48CF-BCFE-596E15399DA7</setguid>
       <propid>marc606xml</propid>
       <customtype>xml</customtype>
       <purpose>retrieve</purpose>
      </indxsrvprop>
     </based>
    </property>
    <property>
     <name>author</name>
     <based>
      <indxsrvprop>
       <fname>marc700</fname>
       <datatype>(DBTYPE_WSTR|DBTYPE_BYREF)</datatype>
       <setguid>45807EF7-5D36-48CF-BCFE-596E15399DA7</setguid>
       <propid>marc700</propid>
       <customtype>value</customtype>
       <purpose>search retrieve</purpose>
      </indxsrvprop>
     </based>
    </property>
   </properties>
  </catalog>
 </catalogs>
</configuration>
```

Figura 6.5 – Exemplo de uma configuração do componente IndexsrvXWrapper.

A segunda razão prende-se com a própria funcionalidade do componente. Analisando, por exemplo, a configuração da segunda propriedade, de nome “subject”, verifica-se que esta propriedade baseia-se em duas propriedades presentes no Index Server. O componente utiliza a propriedade “marc606txt” para proceder a pesquisas e utiliza a propriedade “marc606xml” para proceder à recolha de resultados. Este foi o método encontrado para conseguir recolher os valores de um elemento XML presente múltiplas vezes no mesmo documento. A informação recebida consiste no conjunto dos elementos repetidos, em XML, contendo os seus valores. Do ponto de vista do Index Server, este envia apenas um valor, uma *string*; do ponto de vista do componente, este recebe XML que depois de interpretado, transforma num conjunto de múltiplos valores.

Uma questão subsiste contudo: porquê utilizar duas propriedades do Index Server para chegar a este resultado? Este artifício tem como único objectivo evitar a possibilidade de os nomes dos próprios elementos XML serem pesquisados. Estes elementos, como se sabe, são delimitadores de informação, não têm por objectivo a pesquisa própria, o que poderia eventualmente acontecer se fosse utilizada apenas a propriedade “marc606xml”. A utilização única da propriedade “marc606txt” está fora de causa, uma vez que essa propriedade fornece acesso aos diferentes valores do elemento repetido, mas sem a devida individualização.

#### 6.3.2.3 A Aplicação de Teste

Com o objectivo de testar o componente IndexsrvXWrapper foi desenvolvida uma pequena aplicação, de nome IndxsrvXWrapperTest, que possui interface com o utilizador, permite efectuar pesquisas através do componente e visualizar os seus resultados. Na Figura 6.6 encontra-se ilustrada essa aplicação.

Na ilustração da Figura 6.6, a aplicação IndxsrvXWrapperTest mostra os resultados de uma pesquisa recente. Tendo sido efectuada uma pesquisa pelo termo “africa” na propriedade “title” e no catálogo “memafrica”, a janela de resultados informa que foram encontrados 92 documentos que cumprem os requisitos da pesquisa e mostra cada título encontrado e em que documento se encontra.

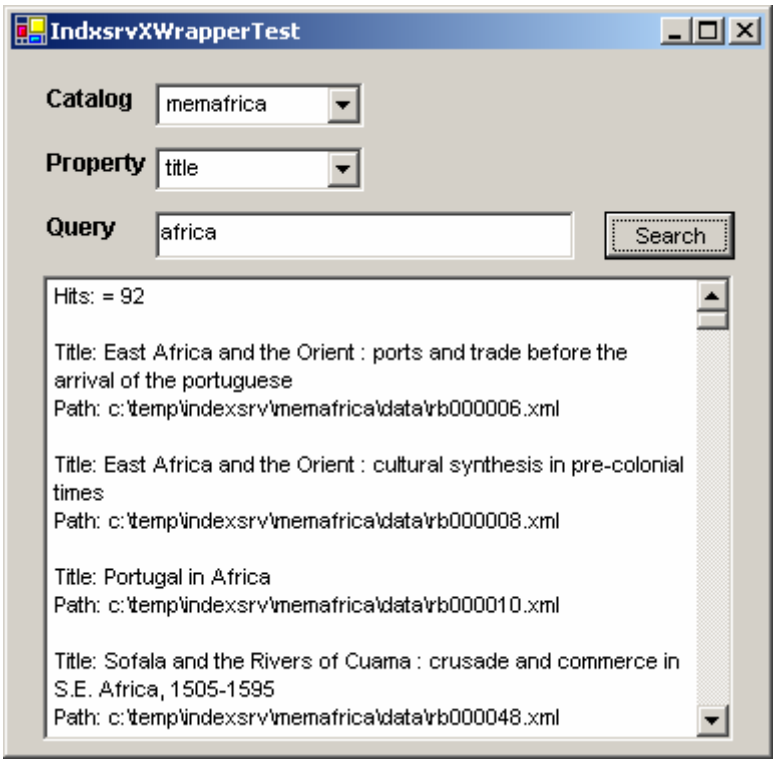



Figura 6.6 – A Aplicação de teste IndxsrvXWrapperTest numa pesquisa.

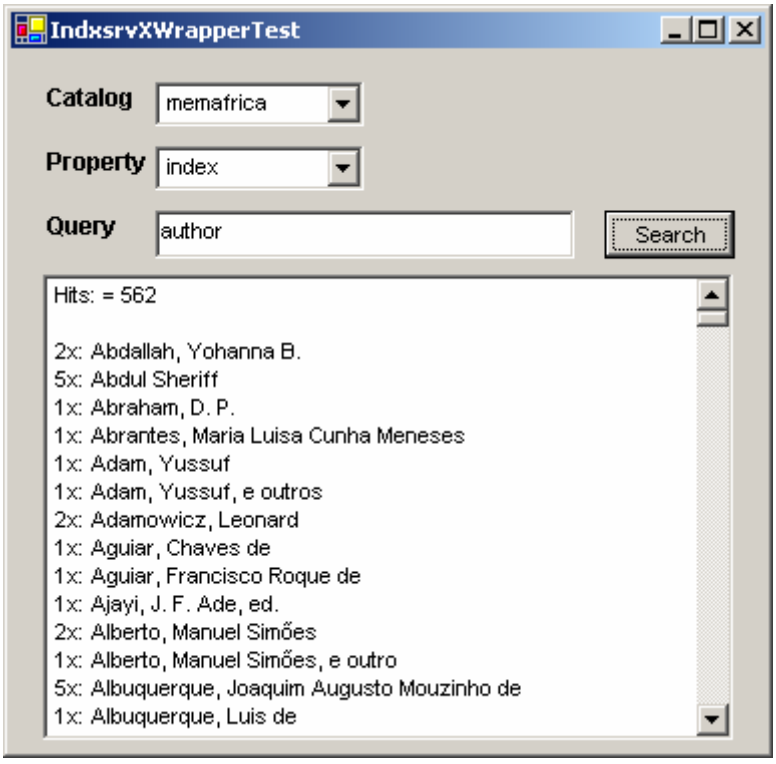



Figura 6.7 – A Aplicação de teste IndxsrvXWrapperTest num pedido de índice.

Na ilustração da Figura 6.7, a aplicação IndxsrvXWrapperTest mostra os resultados de um pedido de índice. Neste caso particular, é pedido o índice da propriedade “author”. O que revela haver 562 nomes de autor diferentes e permite visualizar esses nomes e saber quantas vezes aparecem no conjunto de documentos pertencente ao catálogo “memafrica”.

O conjunto de documentos deste catálogo de teste é composto por 1240 documentos, no formato XML, e consiste numa amostra do conjunto total de registos que compõe o acervo de registos bibliográficos do projecto “Memória de África”. Na Figura 6.8 encontra-se ilustrado um destes registos.

Como é possível averiguar pela visualização do registo ilustrado na Figura 6.8, os elementos XML que o compõem correspondem aos nomes das propriedades constantes no catálogo do Index Server. Contudo, o componente IndexsrvXWrapper, através da sua configuração, traduz esses nomes para nomes mais compreensíveis. O elemento “marc700” transporta o nome do autor; o “marc200” transporta o título e o “marc606”, que aparece repetido, transporta o assunto, etc.. Os nomes destes elementos, embora não se encontrem de acordo com o padrão, representam registos no formato MARC. Neste caso particular trata-se da nuance Unimarc.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<record id="000006" type="unimarc">
<marc700>Chittick, H. Neville</marc700>
<marc200>East Africa and the Orient : ports and trade before the arrival of the
portuguese</marc200>
<marc201>H. Neville Chittick</marc201>
<marc463>Historical relations across the Indian Ocean : report and papers of the
meeting of experts / organized by Unesco (Port Louis, Mauritius, from 15 to 19
July 1974). - Paris : Unesco, 1980. - p. 13-22</marc463>
<marc101>Eng</marc101>
<marc102>FR</marc102>
<marc966>299d|AHM</marc966>
<marc500>East Africa and the Orient</marc500>
<marc606>Moçambique</marc606>
<marc606>História</marc606>
<marc606>Economia</marc606>
<marc606>História económica</marc606>
<marc606>Comércio do Índico</marc606>
<marc606>África oriental</marc606>
<marc606>Asiáticos</marc606>
<marc606>Árabes</marc606>
<marc922>Analítico</marc922>
</record>
```

Figura 6.8 – Registo do acervo do projecto “Memória de África”.

### 6.3.3 O *Web Service* ISFSWS

Por forma a permitir a distribuição da carga e do processamento foi também concebido e desenvolvido um *web service* , o ISFSWS - *Index Server and File System Web Service*, para acesso ao componente IndexsrvXWrapper e ao próprio sistema de ficheiros, onde se encontram armazenados os documentos.

A concepção deste *web service* foi também norteado pelo desejo de contribuir para uma certa homogeneidade das interfaces dos *web services* que permitem o acesso a repositórios de informação, sejam eles bases de dados ou sistemas de ficheiros. Por isso, tentou-se que a sua interface se aproximasse o mais possível daquele que foi desenvolvido para a interface XML:DB. Algumas diferenças na interface são consequência directa das diferentes valências dos dois tipos de repositórios.

Na Figura 6.9 encontra-se a representação do modelo WSDL deste *web service*, tendo sido desenvolvido na plataforma .Net.

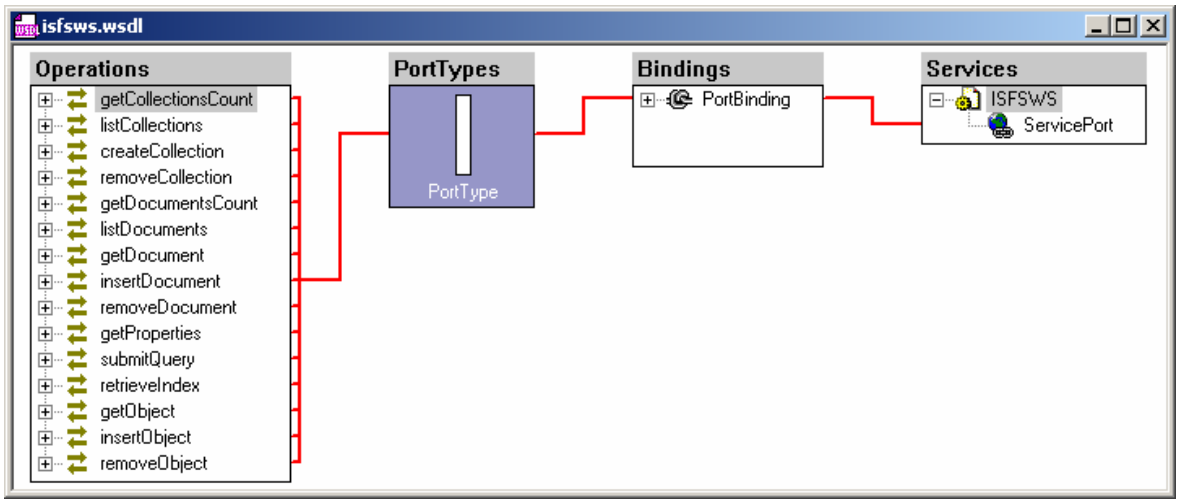



Figura 6.9 – Representação do modelo WSDL do ISFSWS.

Tendo em atenção a representação do seu modelo WSDL, verifica-se, de facto, uma grande proximidade entre a interface deste *web service* e a do XML:DB *web service*. As principais diferenças residem na introdução de mais dois métodos, o “getProperties” e o “retrieveIndex”. Os restantes métodos são o mesmos, tanto em número como no nome.

Em termos de funcionalidade interna, é óbvia a não existência de uma correspondência directa entre as funcionalidades deste *web service* e as do XML:DB *web service*, pois tratam-se de ambientes de actuação muito diferentes: num, um sistema de ficheiros e uma aplicação de recolha de informação; noutro, uma base de dados.

Também a funcionalidade externa, aquela que é visível aos agentes que utilizam o *web service*, possui algumas diferenças. Neste *web service*, as colecções são vistas como os catálogos disponíveis no Index Server, e as subcolecções, ou colecções dentro de colecções, consistem nas pastas do sistema de ficheiros onde residem os documentos.

## 6.4 Revisão

Neste capítulo descreveu-se principalmente a concepção e desenvolvimento do *middleware* para acesso a dois diferentes tipos de repositórios de informação, que instanciam os elementos funcionais SPRIs da terceira camada da arquitectura da Plataforma de *Middleware* de Suporte a Bibliotecas Digitais.

É feita uma tentativa para homogeneizar as interfaces de acesso aos repositórios, utilizando os *web services*. À interface de acesso ao repositório baseado no sistema de ficheiros são adicionados dois métodos a mais. Esta diferença prende-se com as possibilidades oferecidas pelo indexador, Index Server, que não se encontram presentes nem na base de dados, nem na interface XML:DB.

Para garantir a total compatibilidade entre as duas interfaces, podem-se juntar à interface com a base de dados esses mesmos dois métodos, mas atribuindo-lhes, neste caso, um comportamento nulo, isto é, sem efeito. Esta não será, porventura, a melhor solução, contudo julga-se pior solução a procura da homogeneidade através do sacrifício desses dois métodos, o que levaria à incapacidade de acesso a uma funcionalidade muito importante – a de recolher índices – nos repositórios que a podem oferecer.

# Capítulo 7

# Testes e Avaliação

Com o objectivo de tentar avaliar qualitativamente e quantitativamente algumas das questões mais pertinentes para o desempenho dos sistemas projectados e desenvolvidos no âmbito deste trabalho, foram pensados e executados alguns testes para avaliar:

- a interoperabilidade entre diferentes plataformas, utilizando os *web services*;
- o possível impacto da opção pelos *web services*, como solução técnica para a construção da plataforma de *middleware*;
- as capacidades de resposta do Agregador de Registos Bibliográficos, sob o efeito de testes de carga.

Os testes e os resultados obtidos são descritos de seguida.

## 7.1 Testes de Interoperabilidade

A interoperabilidade entre *web services* desenvolvidos em diferentes plataformas tem sido um assunto que tem merecido as maiores atenções, praticamente desde o aparecimento dos próprios *web services*. A razão para tal é óbvia: um dos principais objectivos do aparecimento dos web services foi precisamente tentar colmatar a falta de interoperabilidade entre sistemas heterogéneos. Se essa interoperabilidade não existir,

poderá então considerar-se em muitos casos, que os *web services* falharam o seu propósito.

No presente trabalho foram utilizadas, basicamente, duas plataformas:

- a plataforma JAVA;
- e a plataforma .NET.

Na plataforma JAVA, foram essencialmente desenvolvidos *web services* para o Agregador de Registos Bibliográficos e para a interface XML:DB. Na plataforma .NET, foi desenvolvido o *web service* ISFSWS. Por outro lado, os clientes para esses *web services* foram até ao momento essencialmente desenvolvidos na plataforma .NET.

Apresentam-se de seguida dois cenários diferentes que permitiram testar a tão desejada interoperabilidade.

### 7.1.1 Um Servidor JAVA e um Cliente .NET

Como referido antes, o Agregador de Registos Bibliográficos possui um *web service* desenvolvido na plataforma JAVA. Por forma a expor esse catálogo aos utilizadores, no imediato, foi desenvolvida uma interface ao utilizador, baseada na web e sobre a plataforma .NET. Nesta plataforma, foi desenvolvido um componente, na forma de uma *Class Library* da plataforma .NET, com o nome Z3950WSClient, que tem por missão interagir com o *web service* do agregador.

Este caso representa um cenário de utilização real, e não simulado. Este sítio web foi inicialmente, enquanto novidade, bastante utilizado, sendo-o menos agora. Contudo, o que é de realçar, neste caso, foi a capacidade do componente cliente Z3950WSClient de “consumir” sempre, sem qualquer tipo de obstáculo, todos os serviços oferecidos pelo *web service* do agregador. Por isso, pode dizer-se que, pelo menos neste caso, a interoperabilidade entre as diferentes plataformas é efectiva.

### 7.1.2 Um Servidor .NET e um Cliente JAVA

O cenário, envolvendo um *web service* em .NET e um cliente de *web service* em JAVA, não aparece de forma natural no presente trabalho. Por isso, para testar este cenário específico foi utilizado o *web service* ISFSWS, desenvolvido na plataforma .NET, e um cliente JAVA, desenvolvido especialmente para o efeito.

Pode dizer-se neste caso, que se trata de um cenário de utilização simulado, uma vez que não teve, até ao momento, outra utilidade que a de averiguar a interoperabilidade entre as duas plataformas, mas com papeis trocados, em relação ao cenário anterior.

Os resultados dos testes efectuados, revelaram também uma perfeita sincronia entre o servidor e o cliente, não tendo havido qualquer problema por parte do cliente em consumir todos os serviços oferecidos pelo ISFSWS.

## 7.2 Testes de Impacto dos *Web Services*

Sabendo que os *web services* consistem numa tecnologia em que as operações são implementadas através do envio e recepção de mensagens textuais, facilmente se questiona o custo que isso poderá apresentar face a tecnologias que trocam mensagens binárias. Existem, já há algum tempo, alguns estudos efectuados sobre este assunto, mais especificamente, comparando o comportamento de diversas tecnologias que permitem a distribuição (Tian et al., 2003). Em geral, estes estudos penalizam bastante os *web services*.

Contudo, considera-se que esses estudos se concentram sobretudo na comparação de tempos de execução e *overheads* introduzidos nos tamanhos das mensagens, não tendo em conta que a grande desvantagem dos *web services*, do seu ponto de vista, o facto de utilizarem mensagens puramente textuais, é também a sua grande vantagem, se a perspectiva se deslocar para o lado da interoperabilidade.

Desta forma, e com os testes descritos nesta secção, pretende-se apenas contabilizar o custo da utilização dos *web services* face à não utilização de qualquer tecnologia intermediária. Pensa-se que este procedimento é o mais correcto, no contexto do presente trabalho, uma vez que o que realmente se pretende avaliar é o custo da utilização de uma tecnologia que ofereça boas capacidades de distribuição em simultâneo com excelentes capacidades de interoperabilidade.

### 7.2.1 Metodologia

Tendo em conta o contexto do presente trabalho, achou-se por bem conduzir os testes com a realização de operações sobre documentos e utilizando as tecnologias que estão já a ser usadas, nomeadamente as bases de dados XML nativas e o sistema de ficheiros. Estes testes dão uma maior noção do custo da utilização dos *web services*, uma vez que se enquadram em situações mais próximas da realidade.

Desta forma, os testes foram primeiro realizados com uma base de dados XML nativa, a Xindice, e depois repetidos com o sistema de ficheiros. Estes testes consistiram basicamente na realização de três operações sobre documentos: inserção, recolha e remoção. Estas operações foram repetidas vinte mil vezes em quatro situações diferentes:

- sobre um documento de 1KB, por acesso directo ao API;
- sobre um documento de 1KB, por acesso via *web services*;
- sobre um documento de 10KB, por acesso directo ao API;
- sobre um documento de 10KB, por acesso via *web services*.

Todos os testes foram conduzidos na mesma máquina para não serem contabilizados os tempos de atraso introduzidos pela rede. Estes testes pretendem sobretudo estudar o impacto da introdução de mais uma camada de processamento.

A máquina onde foram realizados os testes, possuía a seguinte configuração:

- CPU = P4 2.4Ghz;
- RAM = 256 MB;
- Disco = 30 GB (4500rpm);
- S.O. = Windows XP Pro;
- JVM = j2re1.4.2_01;
- Suporte aos *web services* = jakarta-tomcat-5.0.12 e axis v1.1.

### 7.2.2 Teste com a Base de Dados Xindice

Os testes com a base de dados Xindice foram sempre executados através da interface XML:DB, via acesso directo ao seu API ou via *web services* (xmldbws).

Na utilização do documento de 1KB, os resultados obtidos encontram-se expostos no gráfico da Figura 6.10; na utilização do documento de 10KB, os resultados encontram-se no gráfico da Figura 6.11.

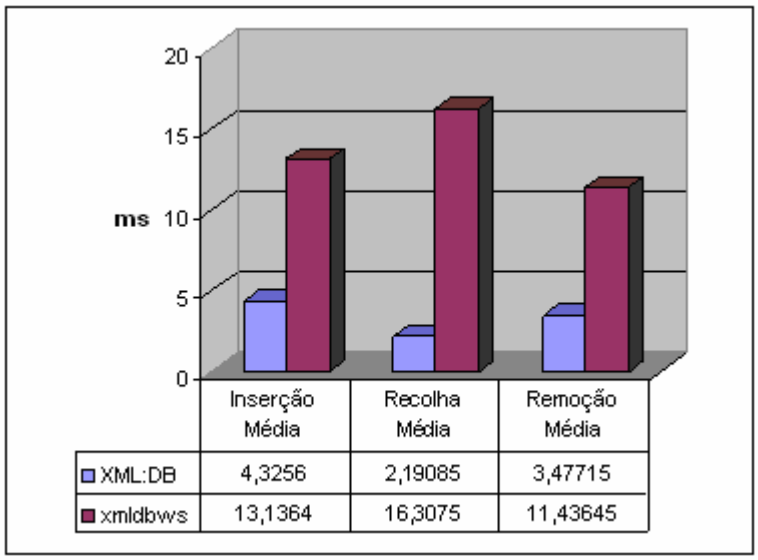



Figura 7.1 – Resultados sobre a base de dados com um documento de 1KB.

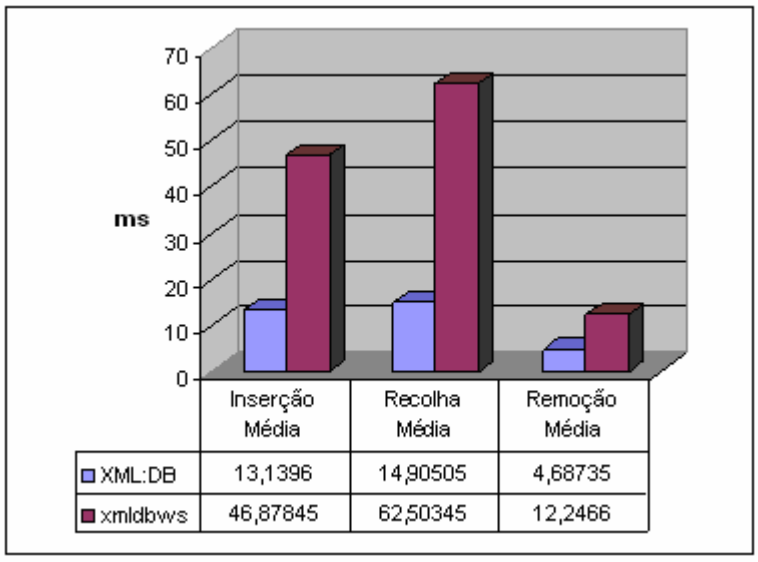



Figura 7.2 – Resultados sobre a base de dados com um documento de 10KB.

Da análise dos resultados obtidos verifica-se que a execução de qualquer das operações, por acesso via *web services*, é penalizada em tempo, o que era de esperar. Para o caso do documento de 1KB, as diferenças percentuais são a seguintes:

- a inserção demora mais 304%;
- a recolha demora mais 744%;
- a remoção demora mais 324%.

Para o caso do documento de 10KB, as diferenças são as seguintes:

- a inserção demora mais 357%;
- a recolha demora mais 419%;
- a remoção demora mais 261%.

### 7.2.3 Teste com o Sistema de Ficheiros

O teste com o sistema de ficheiros utilizou as directivas ao sistema operativo disponíveis na linguagem de programação, sem mais interfaces intermediárias.

Na utilização do documento de 1KB, os resultados obtidos encontram-se expostos no gráfico da Figura 6.12; na utilização do documento de 10KB, os resultados encontram-se no gráfico da Figura 6.13.

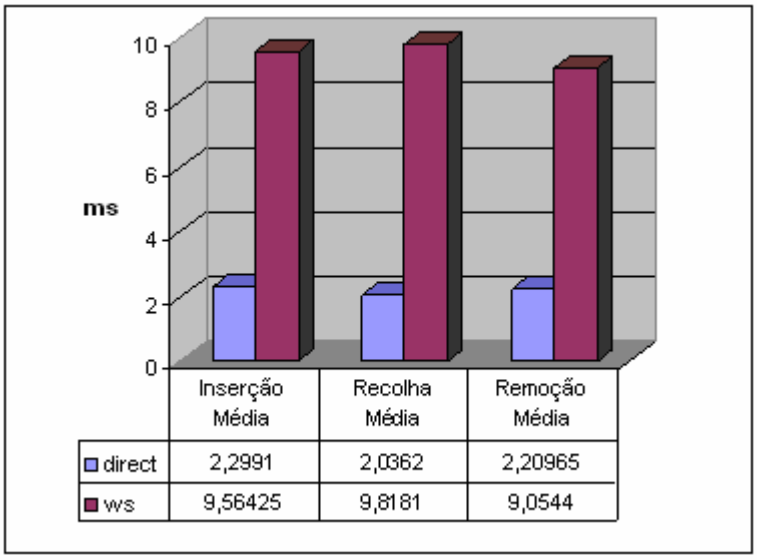



Figura 7.3 – Resultados sobre o sistema de ficheiros com um documento de 1KB.

Mais uma vez, como seria de esperar, os acessos via *web services* são penalizados no tempo. Para o caso do documento de 1KB, as diferenças são as seguintes:

- a inserção demora mais 416%;
- a recolha demora mais 482%;
- a remoção demora mais 410%.

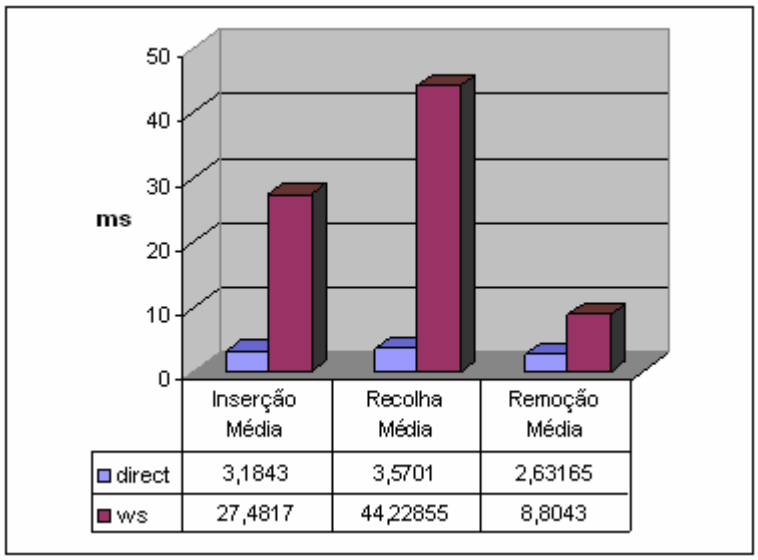



Figura 7.4 – Resultados sobre o sistema de ficheiros com um documento de 10KB.

Para o caso do documento de 10KB, as diferenças são as seguintes:

- a inserção demora mais 863%;
- a recolha demora mais 1239%;
- a remoção demora mais 335%.

### 7.2.4 Algumas Reflexões sobre os Resultados dos Testes

Analisando os resultados obtidos nos testes efectuados, tanto com a base de dados como com o sistema de ficheiros, verifica-se, numa primeira observação, uma coerência quase perfeita dos resultados. Embora os tempos e as diferenças de tempo sejam diferentes nos diversos cenários, verifica-se um padrão inquestionável entre as próprias operações, senão veja-se, em quase todas as situações:

- a operação de remoção foi a que demorou menos e a qual apresenta uma menor diferença de tempos entre o acesso directo e via *web services*;
- a operação de recolha é a que apresenta maiores tempos e maiores diferenças de tempo;
- a operação de inserção fica entre as anteriores.

Existe apenas um cenário em que o padrão não é seguido: os tempos de recolha e remoção do documento de 1KB, em acesso directo, tanto na base de dados como no

sistema de ficheiros. Contudo, este facto não é relevante para o estudo do impacto da utilização dos *web services*, visto que a utilização destes segue o padrão em todos os cenários.

Uma possível explicação para a excepção referida, prende-se com as operações internas da base de dados e do sistemas de ficheiros, que consistem na necessidade de escrita de informação de sinalização na operação de remoção e na necessidade de leitura de dados do ficheiro na operação de recolha.

A segunda observação vai directamente ao encontro do factor mais penalizador da utilização dos *web services*: o tamanho dos dados a transportar, “personificados” nestes testes pelos documentos de 1KB e de 10KB.

Tanto nos acessos à base de dados como ao sistema de ficheiros, as diferenças entre os tempos de acesso directo e de acesso via *web services* aumentam drasticamente na utilização do documento de 10KB. Excepção feita, na operação de remoção, que de facto não transporta o documento, e acaba por certificar esta mesma assunção. Esta não será, por ventura, uma surpresa, uma vez que uma maior quantidade de dados a transportar implica forçosamente um maior tempo de processamento dos mesmos para a sua codificação conforme ao padrão SOAP.

A terceira observação é de extrema importância, pois está directamente relacionada com o objectivo do estudo destes testes, que é o diferencial em tempo real entre a utilização e a não utilização dos *web services*. Contudo por se encontrar intimamente ligada à segunda observação, pode não se revestir de carácter vinculativo, por existirem múltiplos cenários potenciais que a contrariam. Nos testes efectuados, a maior diferença pautou-se por 4 a 5 dezenas de milissegundos, o que, à primeira vista, não se afigura muito importante e viabiliza perfeitamente a utilização desta tecnologia. Mas, como foi dito antes, este valor está fortemente dependente do tamanho dos dados a transportar, o que impede que esta observação seja conclusiva.

## 7.3 Testes de Carga sobre o Agregador de Registos Bibliográficos

Após o desenvolvimento do sistema, pretendeu-se avaliar as suas reais capacidades, assim como os seus limites. Para isso foi desenvolvido um pequeno programa que permite o envio de pedidos de pesquisa ao módulo SPD, simulando utilizadores, por forma a avaliar o seu desempenho em diferentes situações de carga. Embora o sistema não seja constituído apenas pelo módulo SPD, é deste que depende substancialmente.

### 7.3.1 Metodologia

Para a elaboração da metodologia foi tido em conta os diferentes cenários a que o módulo SPD tem de fazer frente. Na verdade, existem duas frentes às quais o módulo tem de fazer face:

- a pesquisa simultânea em múltiplos servidores Z39.50;
- a pesquisa simultânea por parte de múltiplos utilizadores.

Por forma a cruzar estas duas frentes, foram pensadas duas categorias diferentes de testes:

- testes de pesquisa a um único servidor Z39.50;
- testes de pesquisa a todos os servidores Z39.50 simultaneamente.

Para cada categoria, os testes iniciaram com uma única pesquisa, aumentando depois progressivamente o número de pesquisas simultâneas (múltiplos utilizadores simultâneos) até valores que se pensaram ser suficientemente demonstrativos.

A pesquisa que serviu de base a todos os testes foi única e consistia na procura da palavra “java” no campo “título”.

Os parâmetros medidos nos testes foram:

- o tempo de satisfação da pesquisa;
- o tempo de execução da sessão ou sessões aos servidores Z39.50;
- o tempo total de pesquisa;
- o número total de pesquisas simultâneas.

O tempo de satisfação de uma pesquisa é o tempo que medeia entre o início da pesquisa e a satisfação do seu pedido em número de registos. Este parâmetro é muito importante porque, do ponto de vista do utilizador, este pode ser visto como o tempo de resposta do sistema a um pedido de pesquisa, apesar de a pesquisa não estar concluída.

O tempo de execução da sessão ou sessões é diferente, dependendo da categoria dos testes. Na primeira categoria apenas há uma sessão por pesquisa e por isso trata-se do tempo entre o início e o fim dessa sessão. Na segunda categoria, em que há múltiplas sessões para cada pesquisa, é o tempo entre o início da primeira sessão a entrar em execução e o fim da última sessão a sair de execução.

O tempo total de pesquisa é o tempo que medeia entre o início da pesquisa e a sua conclusão.

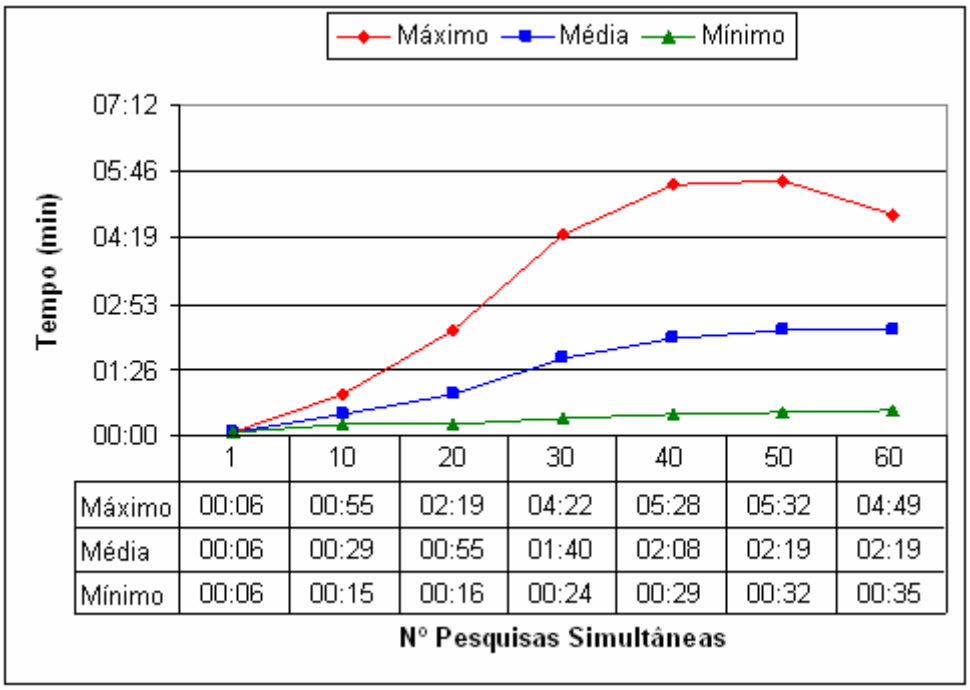
Na configuração dos servidores, para os testes, foram utilizados valores que se julgaram aproximados das situações reais de pesquisa por parte dos utilizadores: limite de 10 registos para a satisfação da pesquisa e 50 registos para a conclusão da sessão com o servidor. Isto quer dizer que para ambas as categorias de testes, ao fim de 10 registos as pesquisas eram dadas como satisfeitas, mas na conclusão das pesquisas, o número máximo de registos poderia ir até aos 50 para a primeira categoria e 450 (50 registos × 9 servidores) para a segunda categoria.

A configuração da máquina foi a mesma que a que foi apresentada na metodologia dos testes de impacto dos *web services*.

### 7.3.2 Pesquisas a um Servidor

Para a execução dos testes de pesquisa a um único servidor, foi escolhido o servidor da Universidade de Brunel. Este servidor tinha já anteriormente mostrado, em alguns testes, ser um dos mais rápidos na resposta.

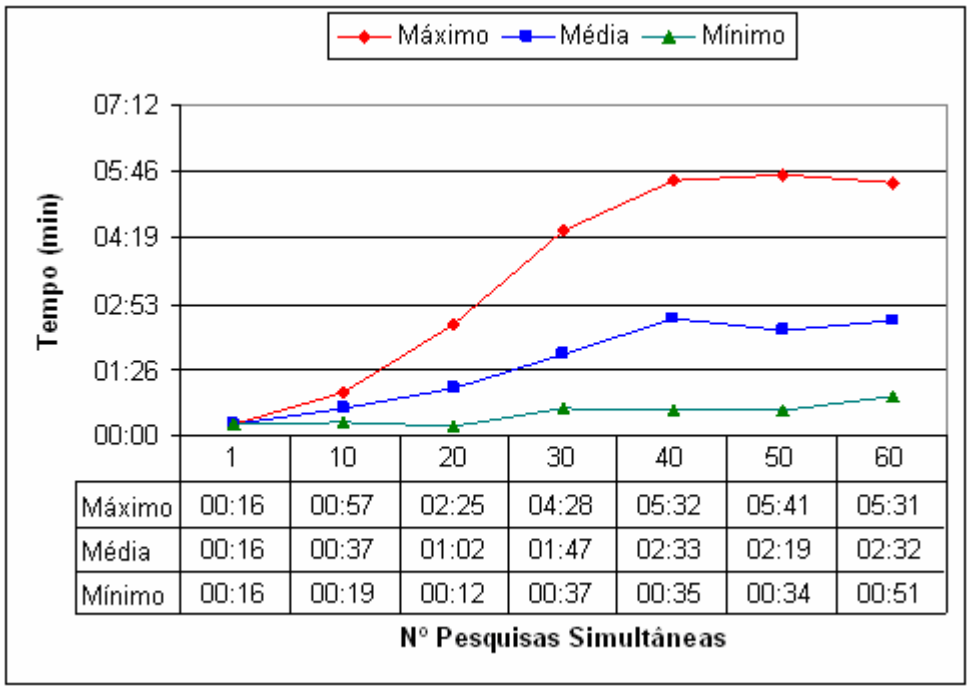
A opção por um servidor rápido esteve unicamente relacionada com a rapidez de execução dos testes. Apesar de poder influenciar as medições dos tempos de resposta do sistema, não influenciaria a avaliação, que pretendia apenas fazer comparações entre os tempos com diferentes cargas.




| | 1 | 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|---|---|
| Máximo | 00:06 | 00:55 | 02:19 | 04:22 | 05:28 | 05:32 | 04:49 |
| Média | 00:06 | 00:29 | 00:55 | 01:40 | 02:08 | 02:19 | 02:19 |
| Mínimo | 00:06 | 00:15 | 00:16 | 00:24 | 00:29 | 00:32 | 00:35 |

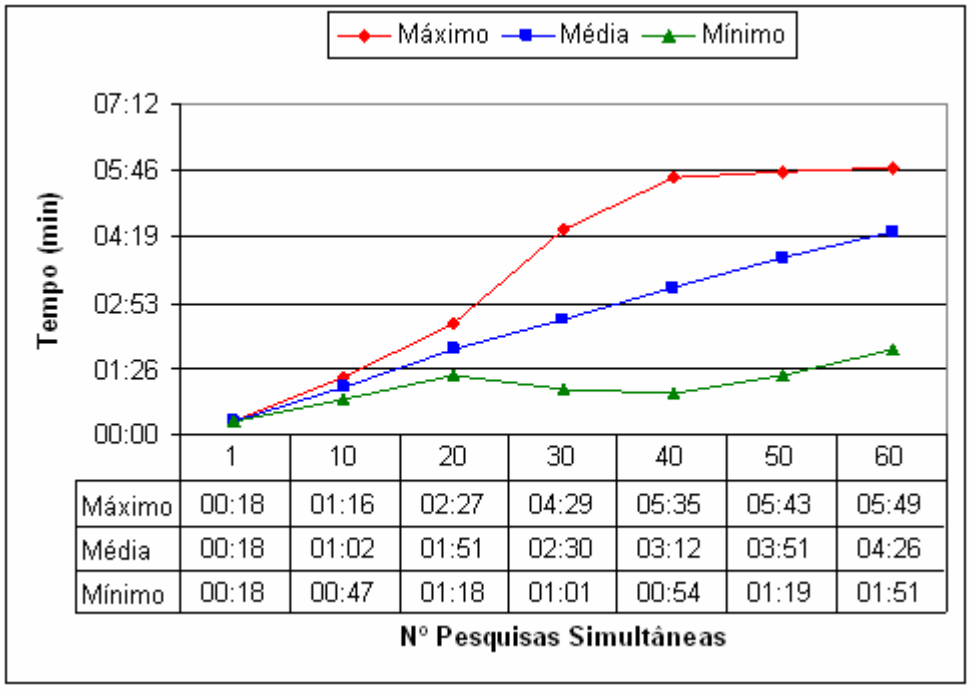
Figura 7.5 – Gráfico e valores dos tempos na satisfação da pesquisa.

A realização dos testes foi sempre efectuada durante a noite, após a meia-noite, para tentar evitar um eventual congestionamento do servidor, que fosse posteriormente mal interpretado como um "ataque" ao mesmo.




| | 1 | 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|---|---|
| Máximo | 00:16 | 00:57 | 02:25 | 04:28 | 05:32 | 05:41 | 05:31 |
| Média | 00:16 | 00:37 | 01:02 | 01:47 | 02:33 | 02:19 | 02:32 |
| Mínimo | 00:16 | 00:19 | 00:12 | 00:37 | 00:35 | 00:34 | 00:51 |

Figura 7.6 – Gráfico e valores dos tempos de execução das sessões.




| | 1 | 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|---|---|
| Máximo | 00:18 | 01:16 | 02:27 | 04:29 | 05:35 | 05:43 | 05:49 |
| Média | 00:18 | 01:02 | 01:51 | 02:30 | 03:12 | 03:51 | 04:26 |
| Mínimo | 00:18 | 00:47 | 01:18 | 01:01 | 00:54 | 01:19 | 01:51 |

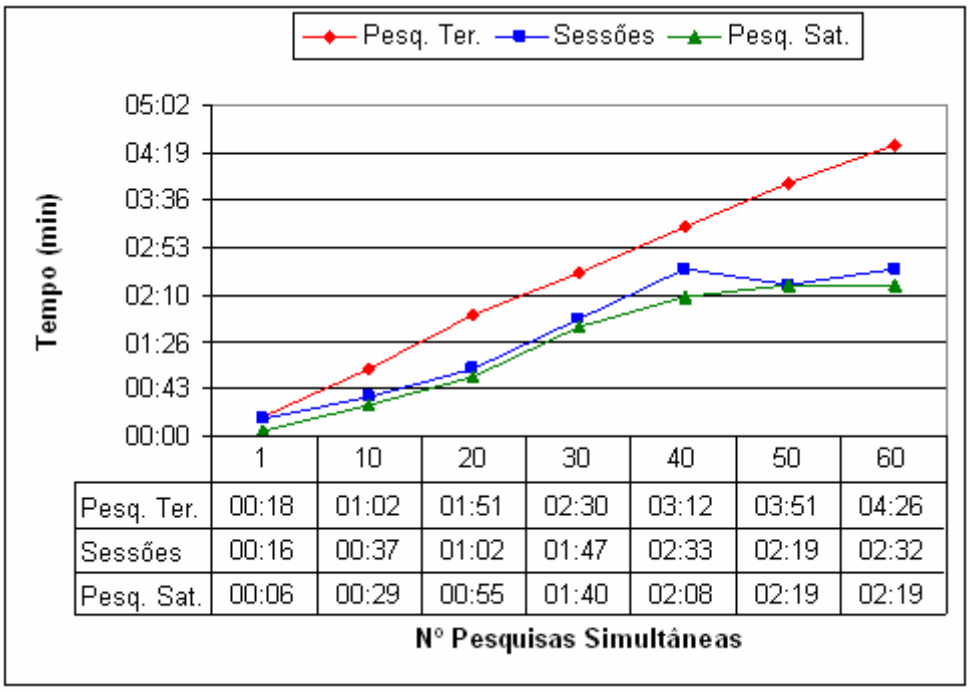
Figura 7.7 – Gráfico e valores dos tempos de execução total das pesquisas.

Nas figuras Figura 7.5, Figura 7.6 e Figura 7.7 encontram-se presentes os valores obtidos das medições efectuadas, bem assim como os seus gráficos.

Da análise dos valores e gráficos, verifica-se que até às 40 pesquisas simultâneas há uma certa linearidade na progressão dos tempos. Após esse valor verifica-se alguma estagnação e até alguma redução, no caso dos valores máximos.

A explicação para estes valores deve-se ao facto de, a partir das 40 pesquisas simultâneas, haver pesquisas que não chegam sequer ao seu ponto de satisfação, terminando antes. De facto, para além deste valor, um número de pesquisas proporcional ao excedente terminam sem atingir o seu objectivo, isto é, não recebem qualquer registo ou recebem um número inferior ao valor de satisfação. Insistindo e aumentando o número de pesquisas simultâneas para além das 50, leva a uma redução drástica das pesquisas que atingem o objectivo e a um aumento dos seus tempos de execução.

Relacionando agora os tempos dos vários quadros é possível produzir o gráfico da Figura 7.8 que apresenta uma relação entre os vários tempos médios: o tempo de satisfação das pesquisas, o tempo de execução das sessões e o tempo de terminação das pesquisas. Este gráfico é principalmente revelador no que toca à razão dos elevados tempos nos términos das pesquisas.



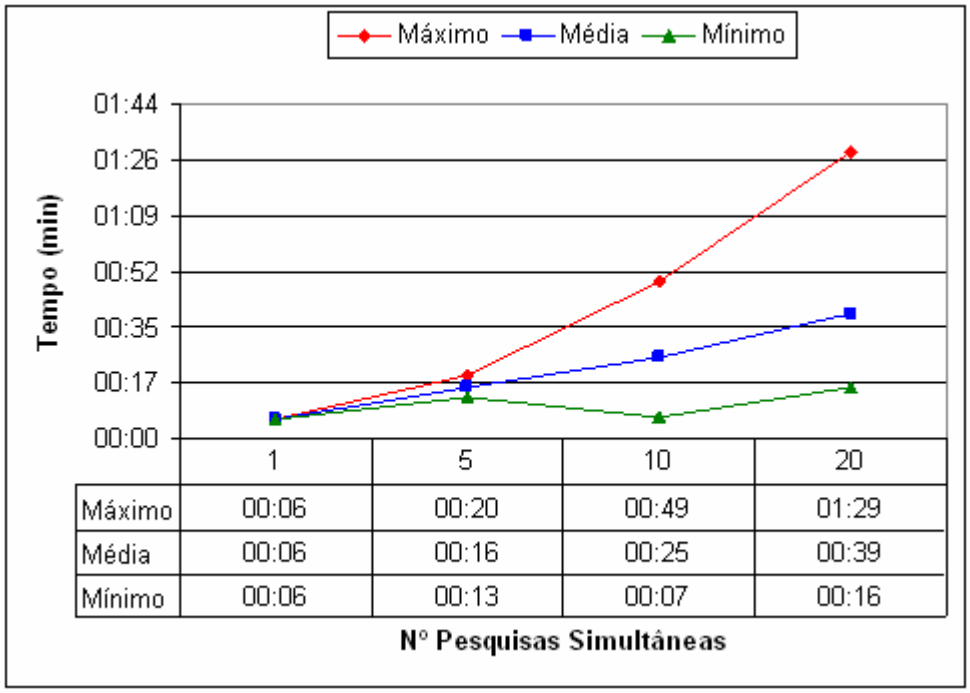

| | 1 | 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|---|---|
| Pesq. Ter. | 00:18 | 01:02 | 01:51 | 02:30 | 03:12 | 03:51 | 04:26 |
| Sessões | 00:16 | 00:37 | 01:02 | 01:47 | 02:33 | 02:19 | 02:32 |
| Pesq. Sat. | 00:06 | 00:29 | 00:55 | 01:40 | 02:08 | 02:19 | 02:19 |

Figura 7.8 – Gráfico de relação entre os diversos tempos médios.

Olhando com atenção, verifica-se que a linha do tempo médio de execução das sessões segue bem junto à linha do tempo médio de satisfação das pesquisas, deixando a linha do tempo médio de término das pesquisas algo afastada. Este facto revela que o tempo necessário para a recolha dos registos é bastante inferior ao tempo necessário para a conclusão da execução das pesquisas, o que implica directamente um elevado tempo para o processamento dos registos recebidos.

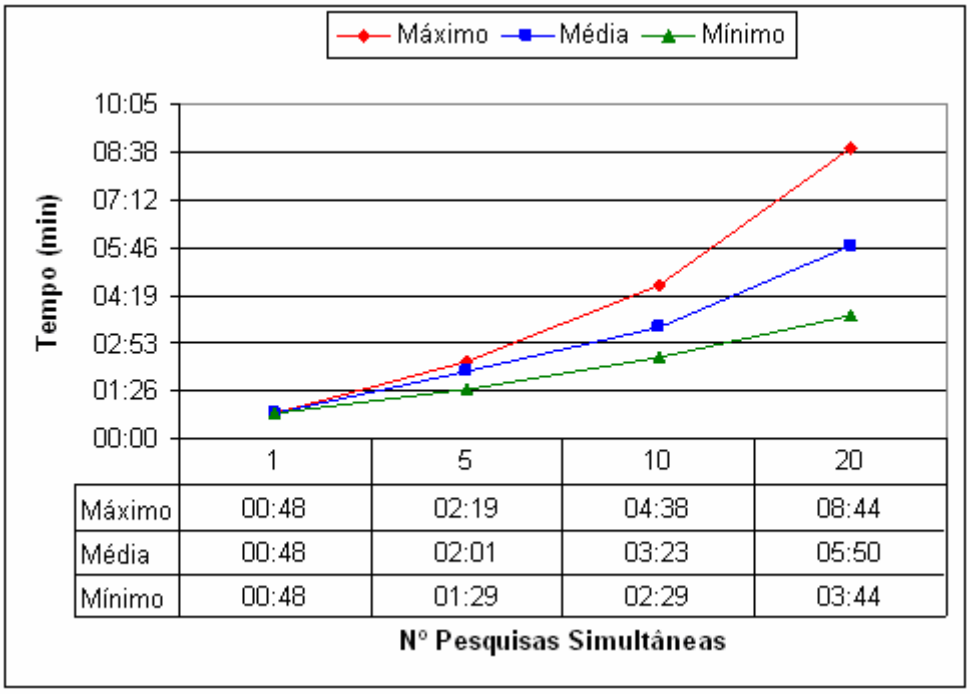
### 7.3.3 Pesquisas a todos os Servidores

Os servidores utilizados para os testes desta categoria são os mesmos que os visualizados na página de entrada da interface de utilizador web desenvolvida para o catálogo colectivo virtual. Também neste caso, os testes foram realizados durante a noite, apesar de não terem sido tão "agressivos" para os servidores, uma vez que o número de pesquisas simultâneas ficou-se por um número bastante abaixo do verificado nos testes anteriores.



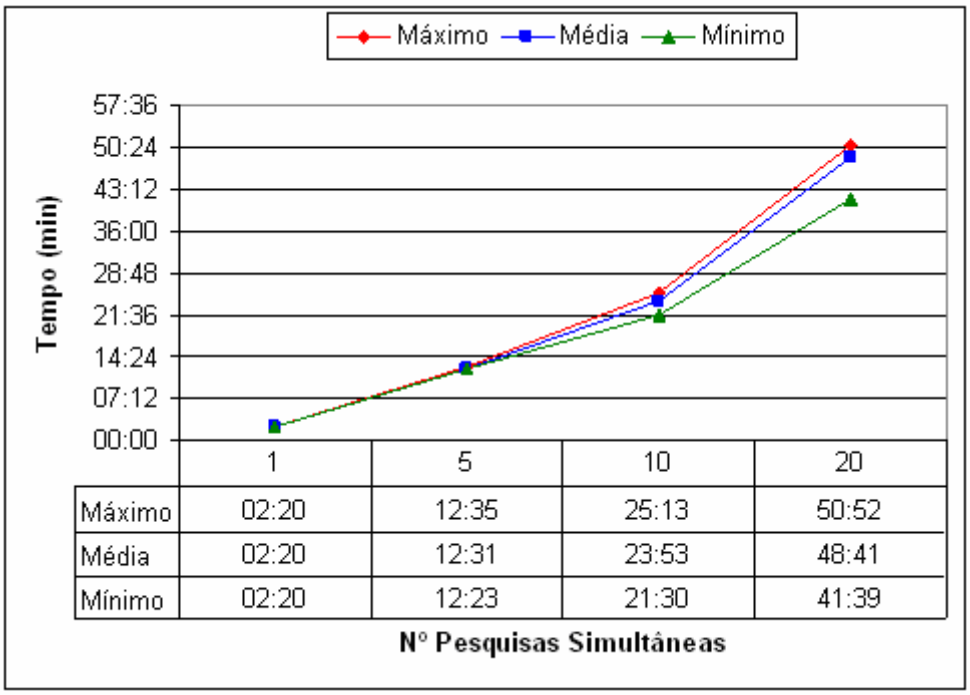

| | 1 | 5 | 10 | 20 |
|---|---|---|---|---|
| Máximo | 00:06 | 00:20 | 00:49 | 01:29 |
| Média | 00:06 | 00:16 | 00:25 | 00:39 |
| Mínimo | 00:06 | 00:13 | 00:07 | 00:16 |

Figura 7.9 – Gráfico e valores dos tempos na satisfação da pesquisa.




| | 1 | 5 | 10 | 20 |
|---|---|---|---|---|
| Máximo | 00:48 | 02:19 | 04:38 | 08:44 |
| Média | 00:48 | 02:01 | 03:23 | 05:50 |
| Mínimo | 00:48 | 01:29 | 02:29 | 03:44 |

Figura 7.10 – Gráfico e valores dos tempos de execução das sessões.




| | 1 | 5 | 10 | 20 |
|---|---|---|---|---|
| Máximo | 02:20 | 12:35 | 25:13 | 50:52 |
| Média | 02:20 | 12:31 | 23:53 | 48:41 |
| Mínimo | 02:20 | 12:23 | 21:30 | 41:39 |

Figura 7.11 – Gráfico e valores dos tempos de execução total das pesquisas.

Os testes realizados nesta categoria ficaram-se por um número de pesquisas simultâneas bastante inferior aos da categoria anterior. A razão para tal prende-se com o tempo necessário à execução das pesquisas. Como é visível na Figura 7.11, o tempo total de execução de 20 pesquisas simultâneas abeira-se de 1 hora. Embora não tenha sido detectada qualquer limitação de execução ao número de pesquisas simultâneas até este valor, foi decidido que não faria sentido a prorrogação dos testes até valores superiores uma vez que se encontra já espelhado um dos factores mais limitativos na utilização do sistema: o tempo.

Os valores patentes na Figura 7.9 revelam contudo que o sistema pode ser usável até a este número de pesquisas simultâneas. Em média, o utilizador obtém uma resposta do sistema até aos 40 segundos. Claro que após essa primeira resposta não se encontram de imediato disponíveis os restantes resultados, o que será certamente motivo de frustração, principalmente se necessitar de esperar mais 40 segundos, em média, para obter uma nova página de resultados.

Os valores da Figura 7.10 revelam também tempos de sessão elevados, quase 6 minutos, em média, contudo ficam bastante longe dos tempos de 50 minutos da Figura 7.11.

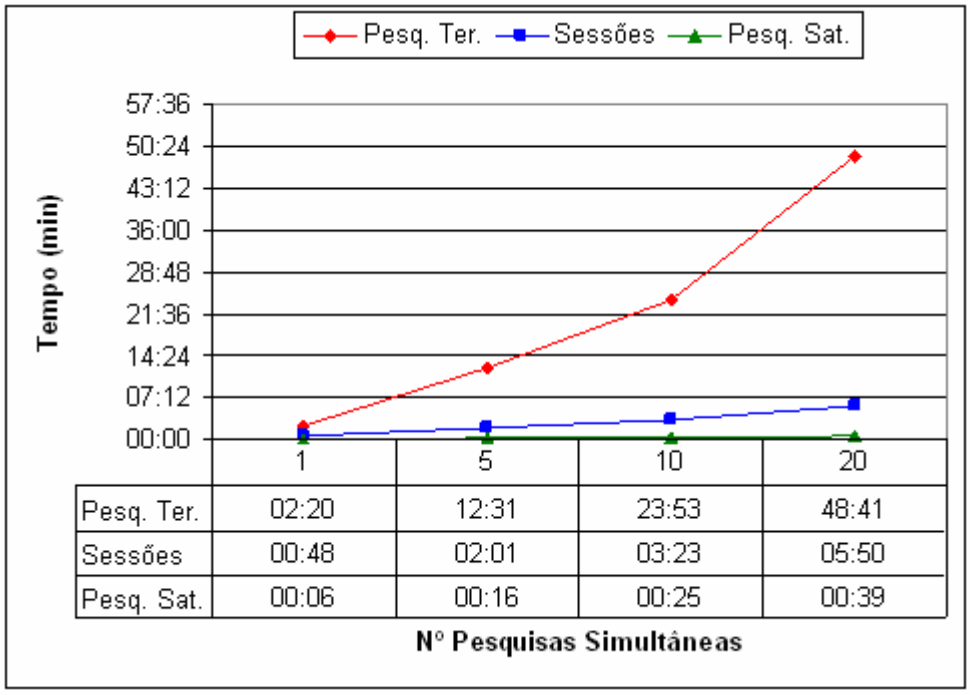



Figura 7.12 – Gráfico de relação entre os diversos tempos médios.

À imagem da Figura 7.8, a Figura 7.12 mostra também uma relação entre os diversos tempos médios obtidos nos testes desta categoria. Esta figura, mais não faz que confirmar de uma forma absoluta a razão para tão grandes esperas na conclusão das pesquisas. A razão é o processamento dos registos na máquina e não a recolha dos registos nos servidores, como fica patente na diferença de 5:50 min para 48:41 min., respectivamente o tempo médio de duração das sessões e o tempo médio de término das pesquisas.

### 7.3.4 Algumas Reflexões sobre os Resultados dos Testes

Na primeira categoria de testes, “pesquisas a um servidor”, verificou-se que o número de pesquisas simultâneas não deveria ascender a mais de 40, sob pena de não se conseguir sequer o processamento desse número de pesquisas. Isto para uma máquina com as especificações descritas anteriormente. Para outra máquina, o número será certamente diferente.

O tempo médio de satisfação de uma pesquisa, para 40 pesquisas simultâneas, é superior a 2 minutos, o que se traduz numa baixa taxa de utilização. Se se pretenderem tempos até 1 minuto, em média, então o valor máximo de pesquisas simultâneas cai para 20. Não se pode esquecer que o tempo de satisfação da pesquisa é o tempo que o utilizador tem de esperar desde que inicia a pesquisa até visualizar a primeira página de resultados.

Na segunda categoria de testes, “pesquisas a todos os servidores”, o número de pesquisas simultâneas ficou-se pelas 20. Este valor não pode ser considerado um limite à execução de mais pesquisas simultâneas, contudo é um valor para além do qual não se apresenta interesse em prosseguir, devido aos tempos elevados apresentados na concretização das pesquisas.

O tempo médio de satisfação das pesquisas, neste cenário, com 20 pesquisas simultâneas, é inferior a 1 minuto, o que pode ser considerado satisfatório, contudo a demora da conclusão das pesquisas, que pode ir até quase 1 hora, preconiza um baixo nível de prestação por parte do sistema ao utilizador.

Comparando as duas categorias de testes, encontra-se facilmente a explicação para as diferenças nos números de pesquisas simultâneas e para os tempos alcançados. Na primeira categoria, uma pesquisa pode recolher e processar até um máximo de 50 registos. Para 40 pesquisas simultâneas, o valor total de registos é de 2000 (40 × 50). Na segunda categoria, uma pesquisa pode recolher e processar até um máximo de 450

registos (9 servidores × 50 registos). Para 20 pesquisas simultâneas, o valor total de registos é de 9000 (20 × 450).

Como foi já averiguado anteriormente, a causa para tão elevados tempos na conclusão das pesquisas é o processamento dos registos localmente na máquina. Esse processamento consiste na conversão dos registos e posterior identificação de duplicados. Durante a execução dos testes, foi possível averiguar, utilizando a ferramenta *Task Manager*, disponível no Windows, que o processo da JVM sobre a qual executa a base de dados Xindice, consome entre 80% a 90% dos recursos do processador. Isto é indicador de que é a tarefa de identificação de registos duplicados e de salvaguarda dos registos, com elevado número de acessos e operações na base de dados, que provoca o congestionamento do processamento. A própria base de dados Xindice apresenta diversas limitações, já anteriormente identificadas (Vaidya and Plale, 2003; Almeida, 2004), que contribuem para os elevados tempos das operações em questão.

A solução para este problema tem de obrigatoriamente passar por se conseguir agilizar as operações efectuadas pelo componente “Processador de Duplicados” e este é fortemente dependente de um repositório para a salvaguarda dos registos.

Antevêem-se duas soluções possíveis:

- a utilização de uma base de dados com maior performance, que não penalize tanto as operações de acesso à mesma. O que poderá ser difícil de obter ou pelo menos poderá ser bastante dispendioso;
- ou a opção por uma solução bipartida, com a guarda temporária dos registos em memória, enquanto se processa as operações de identificação e remoção de duplicados, e por fim a sua salvaguarda definitiva na base de dados. O baixo preço da memória actualmente, pode fazer desta opção um bom compromisso.

## 7.4 Revisão

Este capítulo começa por identificar os diversos tipos de informação com que o sistema lida e são apresentadas duas soluções diferentes, em termos de repositórios de informação. Estas soluções foram ainda alvo de vários desenvolvimentos, nomeadamente ao nível da capacidade de distribuição, desenvolvendo *web services* para cada uma delas, e adicionando e melhorando as capacidades de recolha da informação.

Foi apresentado um conjunto de testes e os seus resultados, que servem para tentar avaliar o impacto da utilização da tecnologia XML no nível funcional (*web services*) e ao nível dos dados (base dados Xindice).

Sobre os repositórios de informação, há a dizer, no contexto do presente trabalho, que aqueles que são baseados em sistemas de ficheiros, são mais apropriados para salvaguarda permanente e por isso mais indicados para cenários de arquivo. Apresentam bons índices de desempenho, tanto nas operações descritas nos testes, como em operações de pesquisa por parte do Index Server. Mas apresentam um baixo nível de interactividade com o exterior. Por exemplo, as inserções de novos documentos ou a remoção de documentos existentes, não são reflectidas com suficiente rapidez no catálogo de indexação do Index Server. O que não permite que este tipo de repositório seja usado em cenários de salvaguarda temporária.

Os repositórios baseados em bases de dados XML nativas, são também boa opção para a salvaguarda permanente, uma vez que oferecem boas interfaces para a salvaguarda, recolha e pesquisa de documentos XML. O seu nível de interactividade com o exterior também é bom, visto as alterações às suas colecções se tornarem imediatamente visíveis. Mas apesar desta sua última característica, a sua utilização para a salvaguarda temporária, principalmente quando submetida a carga operacional elevada, pode-se revelar uma decepção. De facto, julga-se que o processamento da estrutura hierárquica do XML é o grande responsável pela morosidade das operações neste tipo de base de dados. Contudo, esta apreciação resulta dos testes efectuados, neste trabalho, a uma única bases de dados deste tipo. O que não invalida que outras bases de dados possam ter desempenhos muito melhores.

A diferenciação entre salvaguarda temporária e permanente, surge essencialmente devido a duas condições diferentes que se verificam no presente trabalho:

- a salvaguarda temporária é aquela que é necessária para suporte ao tratamento dos resultados das pesquisas e que por isso armazena informação com uma validade temporal bastante limitada;
- a salvaguarda permanente presta-se ao armazenamento da informação, de forma continuada no tempo. São os repositórios onde reside de facto a informação a pesquisar.

# Capítulo 8

# Conclusões Finais

## 8.1 Resumo

A evolução tecnológica actual levou a um aumento exponencial do processamento de informação, através da criação de redes que possibilitam a interconexão de todo o tipo de sistemas e de dados diferenciados. Este facto teve grande impacto nos utilizadores, que subitamente se têm vindo a sentir “soterrados” com enormes quantidades de informação. Assim, este impacto tecnológico criou uma clara necessidade, na actual sociedade de informação: a possibilidade de acesso à informação, de um modo o mais holístico possível.

Este trabalho estabeleceu então como meta, a proposta de uma solução de natureza informática para redimir os problemas que surgem no acesso a sistemas e dados heterogéneos, com origem em bibliotecas digitais multifacetadas e distribuídas. Ou seja, capaz de fornecer interoperabilidade entre essas bibliotecas e oferecer aos utilizadores uma perspectiva homogénea sobre essas diversas fontes de informação.

A abordagem para atingir esta meta consistiu na elaboração de um modelo e de uma arquitectura para uma plataforma de *middleware* capaz de suportar a criação de bibliotecas digitais distribuídas.

Assim, e nesta dissertação, começa-se inicialmente por se proceder à exposição do estado da arte no domínio das bibliotecas digitais, apresentando os conceitos

fundamentais que presidem à definição, caracterização e modelação das bibliotecas digitais, assim como a descrição de alguns dos mais relevantes projectos de investigação, modelos, plataformas, protocolos e normas mas utilizadas neste domínio.

De seguida apresentam-se o modelo e arquitectura para a plataforma de *middleware* proposta, apontando tecnologias específicas para o aumento da interoperabilidade e permitir a integração de sistemas e conteúdos. O que leva à apresentação de um caso de estudo – um agregador de registos bibliográficos – como um protótipo e demonstrador de uma instância da plataforma, descrevendo-se a sua concepção e implementação.

São também tecidas algumas considerações sobre os diferentes tipos de repositórios digitais a serem utilizados na plataforma e descreve-se a implementação de *middleware* específico para contribuir para o aumento da interoperabilidade no acesso a esses repositórios.

Por fim, são descritos alguns testes efectuados sobre alguns elementos da plataforma, com o objectivo de avaliar o impacto da tomada de decisão em optar por tecnologias baseados no XML para as diversas soluções.

## 8.2 Contribuições e Conclusões

O presente trabalho, desenvolvido no domínio da investigação das bibliotecas digitais, pretende ter alcançado os objectivos a que se propôs inicialmente, principalmente no respeitante:

- ao aumento da interoperabilidade entre sistemas de informação heterogéneos;
- à integração do acesso a esses sistemas, sendo eles distribuídos;
- ao propiciar uma visão integrada dos diferentes modelos de metadados;
- e à exploração de meios alternativos de armazenamento e indexação da informação.

### 8.2.1 Contribuições

No processo para atingir os objectivos descritos acima, foram desenvolvidos vários esforços de estudo, concepção, aplicação e desenvolvimento de técnicas de importante relevância neste domínio, que redundaram em diversos contributos e se passam a referir:

- concepção de uma arquitectura e plataforma de serviços de *middleware* para o suporte à concretização de uma biblioteca digital federal;

- concepção de um modelo de abstracção genérico para suporte à concepção da arquitectura referida antes, tendo por base principal os conceitos de distribuição dos serviços, paralelismo na execução dos mesmos e recursividade da funcionalidade atribuída a cada um deles;
- estudo e aplicação de técnicas para a normalização de metadados provindos de diferentes formatos ou modelos para o modelo Dublin Core;
- estudo e aplicação de técnicas para a identificação e remoção de registos de metadados duplicados, com base no modelo de metadados Dublin Core;
- estudo e aplicação da tecnologia UDDI para permitir o registo e descoberta de serviços;
- concepção de uma interface funcional comum, de nome SpriInterface, para a caracterização funcional de todos os serviços;
- estudo e aplicação da tecnologia *web services* para a implementação dos serviços;
- concepção de um modelo de dados, baseado em documentos XML, para a implementação de mensagens de pedido e de resposta a serem utilizadas pelos serviços;
- concepção e implementação de um agregador de registos bibliográficos, seguindo a arquitectura proposta para a plataforma de *middleware*, mas utilizando o protocolo Z39.50 para a pesquisa e recolha de informação;
- concepção e implementação de uma interface web para utilização directa do agregador de registos bibliográficos por utilizadores;
- concepção e implementação de uma interface *web service* para exposição da interface XML:DB na plataforma de *middleware*;
- concepção e implementação de um componente de software, chamado IndexsrvXWrapper, para que, em conjunto com o Index Server da Microsoft e um filtro apropriado para este, seja possível a indexação completa da informação existente em documentos XML;
- implementação de testes para a avaliação do nível de interoperabilidade entre *web services* oriundos de plataformas de software diferentes, nomeadamente a JAVA e a .NET;
- implementação de testes para avaliar do impacto da utilização dos *web services*;
- e implementação de testes de carga ao agregador de registos bibliográficos para tentar avaliar o seu desempenho em funcionamento.

### 8.2.2 Conclusões

O desenvolvimento do agregador de registos bibliográficos revelou que tal sistema é passível de ser implementado, apesar dos desafios de engenharia colocados à partida, e que a utilização das tecnologias e padrões abertos actuais contribuem enormemente para interoperabilidade de sistemas heterogéneos, dando como exemplo, os *web services*.

A utilização do Dublin Core, como formato de metadados para a normalização dos formatos dos registos, foi também considerada um êxito, visto não só se ter revelado possível como a sua simplicidade acabou por trazer ao sistema vantagens de ordem processual e vantagens ao utilizador final, por uma melhor compreensão da informação recolhida.

Sobre os repositórios de informação, foi dada uma contribuição para se tentar homogeneizar as interfaces de acesso aos mesmos e aumentar a sua interoperabilidade através da implementação de *web services* para as mesmas. Para isso foi utilizada a interface XML:DB como base, à qual foi apenas necessário adicionar dois métodos para poder oferecer todas as capacidades do sistema de ficheiros indexado pelo Index Server.

Em função dos testes realizados e descritos no capítulo sete, conclui-se que os repositórios baseados em sistemas de ficheiros, são mais apropriados para salvaguarda permanente e por isso mais indicados para cenários de arquivo. Os repositórios baseados em bases de dados XML nativas, são também uma boa opção para a salvaguarda permanente e podem também ser utilizadas para salvaguarda temporária desde que ofereçam boas performances operativas, ao contrário do que acontece com a base de dados Xindice.

Ainda com base nos testes realizados, é possível afirmar que a interoperabilidade através dos *web services* está assegurada e que o impacto do seu uso, em termos de atraso temporal, não é significativo: principalmente em cenários em que o destino da informação são agentes humanos.

## 8.3 Trabalho Futuro

Actualmente, a plataforma Europeana, referida anteriormente, segue um modelo que se aproxima bastante do modelo da plataforma apresentada nesta trabalho. Ambas pretendem constituir pontos únicos de procura de informação que se encontra residente em múltiplos repositórios digitais distribuídos. Ambas funcionam através de sistemas agregadores que se ligam às fontes de informação. As principais diferenças residem no facto de a Europeana utilizar os agregadores para manter um catálogo centralizado de

metadados – onde é feita a pesquisa pedida à plataforma – e no facto de disponibilizar apenas informação sumária acerca de um determinado objecto digital e delegar o acesso ao mesmo na própria fonte onde este se encontra.

Actualmente, a abordagem seguida pela Europena é considerada a mais adequada, visto continuarem existir diferenças de performance muito grandes entre a utilização de catálogos centralizados e catálogos distribuídos. Contudo, sabe-se que essas diferenças de performance são devidas, sobretudo, à utilização de fracos recursos tecnológicos de processamento e de comunicação, tanto do lado do agregador como do lado dos catálogos. Quando os catálogos e os agregadores puderem ter acesso a melhores recursos tecnológicos, que se prevê num curto prazo virem a ter cada vez mais velocidade de acesso e capacidade de processamento, as diferenças detectadas hoje poderão passar a ser tão mínimas que deixarão de ser relevantes. A pensar nesse momento, seria interessante poder-se estudar até que ponto uma plataforma como a Europeana poderia seguir a filosofia integral da plataforma de *middleware* proposta neste projecto de doutoramento. Para tal, essa plataforma teria de obedecer aos seguintes principais critérios:

- a replicação, em tempo real, de todos os pedidos de pesquisa pelos repositórios digitais, em vez da utilização do catálogo centralizado;
- e a possibilidade de mostrar ao utilizador, na própria plataforma, o conteúdo dos objectos digitais, sem delegar essa funcionalidade no repositórios onde estes residem.

Para dar cumprimento a estes critérios, propõe-se a utilização de extensões ao protocolo OAI-PMH, por forma a permitirem-lhe a pesquisa de informação (Mazurek and Werla, 2008; Suleman and Fox, 2001) e a recolha de objectos de digitais (Van de Sompel et al., 2004), visto que originalmente este protocolo não possui essas capacidades.

Ainda com vista à possibilidade de disponibilizar os objectos digitais, com toda a sua riqueza multifacetada, na própria interface da Europeana, propõe-se a utilização do recente protocolo OAI-ORE (Tarrant et al., 2009). Um protocolo que, recorrendo a conceitos do domínio da web semântica, permite a descrição de objectos digitais cujas partes se podem encontrar distribuídas e que dessa forma poderá facilitar a identificação automatizada da informação necessária à apresentação desses objectos.

# Referências


AACR (2006). "Anglo-American Cataloguing Rules." American Library Association, Canadian Library Association, and the Chartered Institute of Library and Information Professionals. <http://www.aacr2.org/> [Dezembro de 2009]

ACM (2009). "ACM Portal - The ACM Digital Library." ACM. <http://portal.acm.org/dl.cfm> [Agosto de 2009]

Almeida, P. (2004). "Servidor de Vídeo de Elevada Granularidade." Universidade de Aveiro, Aveiro.

Almeida, P., Fernandes, M., Alho, M., Martins, J. A. and Pinto, J. S. (2006). "SInBAD - A Digital Library to Aggregate Multimedia Documents". Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services, IEEE Computer Society.

Altova (2009). "Altova XMLSpy 2010 - XML Editor." <http://www.altova.com/xml-editor/> [Dezembro de 2009]

Apache (2006). "Apache Axis." The Apache Web Services Project. <http://ws.apache.org/axis/> [Outubro de 2009]

Apache (2009b). "Apache Lucene." The Apache Software Foundation. <http://lucene.apache.org/> [Dezembro de 2009]

Apache (2009a). "Apache Tomcat." The Apache Software Foundation. <http://tomcat.apache.org/> [Outubro de 2009]

Apache (2007). "Apache Xindice." The Apache XML Project. <http://xml.apache.org/xindice/> [Outubro de 2009]

Arms, W. (2000). "Digital Libraries." The MIT Press, Cambridge, MA.

ArParlamentar (2008). "Arquivo Histórico Parlamentar - Arquivo Audiovisual." <http://av.parlamento.pt> [Dezembro de 2009]

AvDigital (2008). "Programa Aveiro Digital 2003-2006." <http://www.aveiro-digital.pt/> [Dezembro de 2009]

BathGroup (2003). "The Bath Profile: An International Z39.50 Specification for Library Applications and Resource Discovery (Release 2.0)." Library and Archives Canada - Bath Profile Maintenance Agency. <http://collectionscanada.ca/bath/tp-bath2-e.htm> [Janeiro de 2004]

BIBLINK (1999). "Linking Publishers and National Bibliographic Services." <http://cordis.europa.eu/libraries/en/projects/biblink.html> [Agosto de 2009]

BND (2009). "Biblioteca Nacional Digital." Biblioteca Nacional de Portugal. <http://purl.pt/index/geral/PT/index.html> [Dezembro de 2009]

Borbinha, J. (2000). "Bibliotecas Digitais: O Futuro Através da Biblioteca Tradicional." Universidade Técnica de Lisboa - Instituto Superior Técnico, Lisboa.

Borgman, C. (1999). "What are digital libraries? Competing visions." Information Processing Management, vol. 35, no. 3, pp. 227-243

Borgman, C. (2000). "From Gutenberg to the global information infrastructure: access to information in the networked world." MIT Press, Cambridge, MA.

Borgman, C., Bates, M., Cloonan, M., Efthimiadis, E., Gilliland-Swetland, A., Kafai, Y., Leazer, G. and Maddox, A. (1996). "Social aspects of digital libraries: Final report to the National Science Foundation." <http://dli.grainger.uiuc.edu/national.htm> [Julho de 2008]

BRICKS (2007). "BRICKS - Building Resources for Integrated Cultural Knowledge Services." <http://cordis.europa.eu/ist/digicult/bricks.htm> [Setembro de 2009]

Bush, V. (1945). "As We May Think." The Atlantic Monthly. http://www.theatlantic.com/magazine/archive/1969/12/as-we-may-think/3881/ [Julho de 2008]

Bush, V., Nyce, J. M. and Kahn, P. (1991). "From Memex to hypertext : Vannevar Bush and the mind's machine." Academic Press, Boston.

Candela, L., Castelli, D., Ferro, N., Ioannidis, Y., Koutrika, G., Meghini, C., Pagano, P., Ross, S., Soergel, D., Agosti, M., Dobreva, M., Katifori, V. and Schuldt, H. (2007). "The DELOS Digital Library Reference Model - Foundations for Digital Libraries. Version 0.98." <http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf> [Outubro de 2009]

CASA (1999). "Cooperative Archive of Serials and Articles." <http://cordis.europa.eu/libraries/en/projects/casa.html> [Agosto de 2009]

CDSsc (2005). "CDS Invenio." The CERN Document Server Software Consortium. <http://cdsware.cern.ch/invenio/index.html> [Outubro de 2009]

Cleveland, G. (1998). "Digital libraries: Definitions, issues and challenges." UDT Core Programme - IFLANET. <http://archive.ifla.org/VI/5/op/udtop8/udtop8.htm> [Agosto de 2009]

CORBA (2008). "Common Object Request Broker Architecture (CORBA) Specification, Version 3.1." OMG. <http://www.omg.org/spec/CORBA/3.1/Interfaces/PDF> [Setembro de 2009]

Cornell (2000). "Dienst Architecture Summary Description." <http://www.cs.cornell.edu/cdlrg/dienst/architecture/architecture.htm> [Outubro de 2009]

Coyle, K. (2000). "The Virtual Union Catalog: A Comparative Study." D-Lib Magazine, vol. 6, no. 3. <http://www.dlib.org/dlib/march00/coyle/03coyle.html> [Agosto de 2009]

Crossnet (2001). "ZedJava Toolkit." Crossnet Systems. <http://roadrunner.crxnet.com/wwwzedjava.html> [Janeiro de 2001]

Davis, J. and Lagoze, C. (1996). "The Networked Computer Science Technical Report Library - TR96-1595." Cornell University. <http://hdl.handle.net/1813/7250> [Agosto de 2009]

DCMI (2004). "DC-Library Application Profile (DC-Lib)." Dublin Core Metadata Initiative. <http://dublincore.org/documents/library-application-profile/> [Outubro de 2009]

DCMI (2008). "Dublin Core Metadata Element Set, Version 1.1." Dublin Core Metadata Initiative. <http://dublincore.org/documents/dces/> [Janeiro de 2009]

DebParlamentares (2009). "Debates Parlamentares - Catálogos Gerais." <http://debates.parlamento.pt> [Dezembro de 2009]

DELOS (2002). "DELOS - A Network of Excellence on Digital Libraries." <http://delos-noe.iei.pi.cnr.it/> [Setembro de 2009]

DELOS (2007). "DELOS - A Network of Excellence on Digital Libraries." <http://cordis.europa.eu/ist/digicult/delos.htm> [Setembro de 2009]

DL.org (2009). "Coordination Action on Digital Library Interoperability, Best Practices, and Modelling Foundations." <http://cordis.europa.eu/fp7/ict/telearn-digicult/digicult-projects-dlorg_en.html> [Setembro de 2009]

DLconsulting (2009). "Veridian." <http://www.dlconsulting.com/veridian.php> [Outubro de 2009]

DLI1 (1998). "Digital Libraries Iniciative - Phase One." <http://dli.grainger.uiuc.edu/national.htm> [Setembro de 2009]

DLI2 (2004). "Digital Libraries Iniciative - Phase Two." <http://www.nsf.gov/pubs/1998/nsf9863/nsf9863.htm> [Setembro de 2009]

Downes, L. and Mui, C. (2000). "Unleashing the killer app: digital strategies for market dominance." Harvard Business School Press, Boston (MA).

DSpace (2009a). "DSpace 1.5.2 Manual." The DSpace Foundation. <http://www.dspace.org/1_5_2Documentation/DSpace-Manual.pdf> [Agosto de 2009]

DSpace (2009c). "DSpace :: Form Dashboard." <http://www.dspace.org/index.php?option=com_formdashboard&Itemid=151&lang=en> [Julho de 2010]

DSpace (2009b). "DSpace Functional Diagram." <http://www.dspace.org/images/stories/dspace-diagram.pdf> [Agosto de 2009]

DuraSpace (2009). "Fedora Commons and DSpace Foundation Join Together to Create DuraSpace™ Organization." <http://duraspace.org/pressrelease.php> [Setembro de 2009]

ELISE (1995). "Electronic library image service for Europe." <http://cordis.europa.eu/libraries/en/projects/elise.html> [Agosto de 2009]

EPrints (2009). "Open Access and Institutional Repositories with EPrints." University of Southampton, UK. <http://www.eprints.org/> [Outubro de 2009]

ESE (2010b). "About the ESE v3.3 XML Schema." EDL Foundation. <http://version1.europeana.eu/web/guest/technical-requirements/> [Julho de 2010]

ESE (2010a). "Europeana Semantic Elements specifications (version 3.3)." EDL Foundation. <http://version1.europeana.eu/web/guest/technical-requirements/> [Julho de 2010]

EUROPAGATE (1996). "European SR-Z39.50 Gateway." <http://cordis.europa.eu/libraries/en/projects/europaga.html> [Agosto de 2009]

Europeana (2010). "Europeana: Think Culture." EDL Foundation. <http://www.europeana.eu> [Julho de 2010]

Ex_Libris (2008). "MetaLib: Reach Out and Discover Remote Resources." <http://www.exlibrisgroup.com/category/MetaLibOverview> [Dezembro de 2009]

eXist (2009). "eXist-db Open Source Native XML Database." SourceForge.net. <http://exist.sourceforge.net/> [Dezembro de 2009]

Fedora (2009). "Fedora Commons Repository Software." Fedora Commons. <http://fedora-commons.org/> [Outubro de 2009]

Fox, E. (1993). "Sourcebook on Digital Libraries: Report for the National Science Foundation." VPI and SU Computer Science Department, Blacksburg, VA. <http://fox.cs.vt.edu/DLSB.html> [Dezembro de 2009]

FP3 (1994). "Libraries within the third Framework Programme (1990-1994)." <http://cordis.europa.eu/libraries/en/lib-3fp.html> [Agosto de 2009]

FP4 (1998). "Creating a European Library Space Telematics for Libraries Programmes (1990-1998)." <http://cordis.europa.eu/libraries/en/intro.html> [Agosto de 2009]

FP5 (2002). "Digital Heritage & Cultural Content." <http://cordis.europa.eu/ist/ka3/digicult> [Agosto de 2009]

FP6 (2006). "Digital Cultural Heritage." <http://cordis.europa.eu/ist/digicult/index.html> [Agosto de 2009]

FP7 (2007). "Digital Cultural Heritage." <http://cordis.europa.eu/fp7/ict/telearn-digicult/digicult_en.html> [Setembro de 2009]

Gladney, H. M. (2006). "Principles for digital preservation." Communications of ACM, vol. 49, no. 2, pp. 111-116

Gladney, H. M., Fox, E. A., Ahmed, Z., Ashany, R., Belkin, N. J. and Zemankova, M. (1994). "Digital Library: Gross Structure and Requirments (Report from a March 1994 Workshop)". Digital Libraries' 94, Texas, USA.

Goutam, B. and Dibyendu, P. (2010). "An evaluative study on the open source digital library softwares for institutional repository: Special reference to Dspace and greenstone digital library." International Journal of Library and Information Science, vol. 2, no. 1, pp. 1-10. <http://www.academicjournals.org/IJLIS/PDF/pdf2010/Feb/Biswas%20and%20Paul.pdf> [Julho de 2010]

Gutenberg (2009). "Project Gutenberg." <http://www.gutenberg.org/> [Agosto de 2009]

Hamilton, G. (1997). "JavaBeans API Specfication (Version 1.01)". Sun Microsystems.

Hillman, D. (2005). "Using Dublin Core." Dublin Core Metadata Initiative. <http://dublincore.org/documents/usageguide/> [Dezembro de 2009]

HINGO (2009). "Human Info NGO - We Care and Share (Humanitarian Infromation for All)." <http://humaninfo.org> [Agosto de 2009]

HP (2007). "HP Press Release: HP and MIT Create Non-profit Organization to Support Growing Community of DSpace Users." <http://www.hp.com/hpinfo/newsroom/press/2007/070717a.html> [Dezembro de 2009]

i2010DLI (2009). "i2010: Digital Libraries Initiative." European Comission - Information Society. http://ec.europa.eu/information_society/activities/digital_libraries/index_en.htm [Setembro de 2009]

IFLA (1999). "Universal Bibliographic Control and International MARC Core Programme." <http://archive.ifla.org//VI/3/p1996-1/unimarc.htm> [Outubro de 2009]

ILU (2000). "Inter-Language Unification." XEROX. <ftp://ftp.parc.xerox.com/pub/ilu/ilu.html> [Setembro de 2009]

ISO (2008). "ISO 2709:2008 - Information and documentation (Format for information exchange)." <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41319> [Outubro de 2009]

Lagoze, C. and Davis, J. R. (1995a). "Dienst: an architecture for distributed document libraries". vol. 38, no. 4, pp. 47

Lagoze, C., Shaw, E., Davis, J. R. and Krafft, D. B. (1995b). "Dienst: Implementation Reference Manual - TR95-1514." Cornell University. <http://hdl.handle.net/1813/7171> [Dezembro de 2009]

Lagoze, C. and Van de Sompel, H. (2001). "The Open Archives Initiative: Building a low-barrier interoperability framework." First ACM/IEEE Joint Conference on Digital Libraries, Roanoke, VA.

LData (2009). "Linked Data - Connect Distributed Data across the Web." <http://linkeddata.org/> [Outubro de 2009]

Lebert, M. (2008). "Gutenberg Project (1971 - 2008)." <http://www.pg-news.org/20080524/pg-1971-2008-lebert-en/> [Julho de 2009]

Lewis, N. (2002). "Talking about a revolution? First impressions of Ex Libris's MetaLib." Ariadne, no. 32. <http://www.ariadne.ac.uk/issue32/metalib/intro.html> [Julho de 2009]

Licklider, J. (1965). "Libraries of the future." MIT Press, Cambridge, MA.

LoC (1998). "Digital Library Iniciatives - Library of Congress." Library of Congress. <http://lcweb2.loc.gov/ammem/dli2/index.html> [Setembro de 2009]

LoC (2009b). "International Standard Z39.50 Maintenance Agency." The Library of Congress Network Development & MARC Standards Office. <http://www.loc.gov/z3950/agency/> [Outubro de 2009]

LoC (2009e). "Library of Congress Subject Headings, 31st edition (2008-2009)." Cataloging Distribution Service - Bibliographic Products & Services from the Library of Congress. <http://www.loc.gov/cds/lcsh.html> [Outubro de 2009]

LoC (2006). "The MARC 21 Formats: Background and Principles." The Library of Congress Network Development & MARC Standards Office. <http://www.loc.gov/marc/96principl.html> [Outubro de 2009]

LoC (2009f). "MarcXml - Marc 21 XML Schema." <http://www.loc.gov/standards/marcxml/> [Outubro de 2009]

LoC (2009c). "SRU - Search/Retrieval via URL." The Library of Congress Network Development & MARC Standards Office. <http://www.loc.gov/standards/sru/> [Outubro de 2009]

LoC (2009d). "SRW - Search/Retrieve Web Service." The Library of Congress Network Development & MARC Standards Office. <http://www.loc.gov/standards/sru/sru1-1archive/srw.html> [Outubro de 2009]

LoC (2009a). "THOMAS: Legislative Information on the Internet." The Library of Congress. <http://thomas.loc.gov> [Agosto de 2009]

Lunau, C. (1998b). "The Need for an International Z39.50 Profile for Searching Virtual Catalogues". National Library of Canada,

Lunau, C. (1998a). "The Virtual Canadian Union Catalogue Project (vCuc): Using Z39.50 to Emulate a Centralized Union Catalogue". IFLANET - 64th IFLA General Conference, Amsterdam.

Lynch, C. and Garcia-Molina, H. (1995). "Interoperability, scaling, and the digital libraries research agenda: A report on the May 18-19, 1995." IITA Digital Libraries Workshop.

Mazurek, C. and Werla, M. (2008). "Extending OAI-PMH Protocol with Dynamic Sets Definitions using CQL Language". IADIS - International Conference Information Systems, Algarve - Portugal.

MELVYL (2006). "MELVYL - The Catalog of the University of California Libraries." <http://melvyl.cdlib.org/> [Agosto de 2009]

MemAfrica (2009). "Portal das Memórias de África e do Oriente." <http://memoria-africa.ua.pt/> [Dezembro de 2009]

Microsoft (2009a). "Microsoft .NET Framework." <http://www.microsoft.com/net/> [Outubro de 2009]

Microsoft (2009b). "Microsoft ASP.net." <http://www.asp.net/> [Outubro de 2009]

Mohr, G. (2008). "MAGNET v0.1." <http://magnet-uri.sourceforge.net/magnet-draft-overview.txt> [Julho de 2009]

Monnich, M. W. (2001). "KVK - a Meta Catalog of Libraries." LIBER QUARTERLY - The Journal of European Research Libraries, vol. 11, no. 2, pp. 121-127

MORE (1994). "MARC Optical Recognition." <http://cordis.europa.eu/libraries/en/projects/more.html> [Dezembro de 2008]

NISO (1994). "ANSI/NISO Z39.2 - Information Interchange Format." <http://www.niso.org/standards/z39-2-1994R2001/> [Outubro de 2009]

NISO (2006). "Metasearch XML Gateway Implementers Guide - Version 1.0." NISO Metasearch Initiative. <http://www.niso.org/publications/rp/RP-2006-02.pdf> [Outubro de 2009]

NRGL (2009). "Comparison of Selected Software Systems for Creation of Digital Libraries." The National Technical Library in Prague - National Repository of Grey Literature. <http://nrgl.techlib.cz/images/Open_source.pdf> [Dezembro de 2009]

NWG (1985). "RFC 959: FILE TRANSFER PROTOCOL (FTP)." <http://www.ietf.org/rfc/rfc959.txt> [Agosto de 2009]

NWG (1999). "RFC 3986: Hypertext Transfer Protocol -- HTTP/1.1." <http://www.ietf.org/rfc/rfc2616.txt> [Agosto de 2009]

NWG (2005). "The Atom Syndication Format." <http://tools.ietf.org/html/rfc4287> [Outubro de 2009]

NZDL (2009). "The New Zealand Digital Library." The University of Waikato. <http://nzdl.sadl.uleth.ca/cgi-bin/library.cgi> [Agosto de 2009]

NZDLP (2007). "Greenstone." The University of Waikato. <http://www.greenstone.org/> [Outubro de 2009]

OAI (2008b). "The Open Archives Initiative Object Reuse and Exchange." Open Archives Initiative. <http://www.openarchives.org/ore/> [Outubro de 2009]

OAI (2008a). "The Open Archives Initiative Protocol for Metadata Harvesting - Protocol Version 2.0." Open Archives Initiative. <http://www.openarchives.org/OAI/openarchivesprotocol.html> [Outubro de 2009]

OAI (2009). "Registered Data Providers." Open Archives Initiative. <http://www.openarchives.org/Register/BrowseSites> [Outubro de 2009]

OCLC (2009b). "CONTENTdm: Digital Collection Management Software." <http://www.contentdm.org/> [Outubro de 2009]

OCLC (2009a). "The World's Libraries Connected." <http://www.oclc.org/us/en/default.htm> [Outubro de 2009]

OMG (2009). "Documents associated with UML Version 2.2." OMG. <http://www.omg.org/spec/UML/2.2/> [Agosto de 2009]

Paepcke, A. (1996). "Summary of Stanford's Digital Library Testbed Design and Status." D-LIB Magazine. <http://www.dlib.org/dlib/july96/stanford/07paepcke.html> [Dezembro de 2008]

Paepcke, A., Baldonado, M., Chang, C.-C. K., Cousins, S. and Garcia-Molina, H. (2000). "Building the InfoBus: A Review of Technical Choices in the Stanford Digital Library Project." Stanford InfoLab. <http://ilpubs.stanford.edu:8090/472> [Agosto de 2009]

Palvia, S. and Sharma, S. (2007). "E-Government and E-Governance: Definitions/Domain Framework and Status around the World". ICEG'07 - 5th International Conference on E-Governance, Hyderabad, India.

Payette, S. and Rieger, O. (1997). "Z39.50: The User's Perspective." D-Lib Magazine, vol. 3, no. 4. <http://www.dlib.org/dlib/april97/cornell/04payette.html> [Julho de 2004]

Pinto, J., Martins, J., Zagalo, H. and Silva, A. (2000). "Das Bibliotecas Virtuais às Bibliotecas Digitais: Proposta de Arquitectura e Metodologias para Acesso à Informação." INGENIUM - Ordem dos Engenheiros, no. 47, pp. 74-76

Pinto, J. S., Martins, J. A., Almeida, P., Fernandes, M. and Zagalo, H. (2005). "Portuguese Parliamentary Records: A Multimedia Digital Library Distributed Architecture, Based on Web Services". International Conference on Next Generation Web Services Practices (NWeSP'05), Seoul, Korea.

Pirounakis, G. and Nikolaidou, M. (2009). "Comparing Open Source Digital Library Software - Handbook on Digital Libraries." Idea Group Inc. <http://www.dit.hua.gr/~mara/publications/ideaDL09a.pdf> [Dezembro de 2009]

PORBASE (2006). "ZZZ - Serviço de Pesquisa em Servidores Z39.50 Distribuídos." <http://www.porbase.org/produtos/zzz.html> [Outubro de 2009]

QuiLogic (2009). "XML Ifilter for Indexing XML Files." QuiLogic Technologies Inc. <http://quilogic.cc/ifilter.htm> [Dezembro de 2009]

Reich, V. and Winograd, T. (1995). "Working assumptions about the digital library." <http://dbpubs.stanford.edu:8091/diglib/pub/reports> [Julho de 2004]

Reis, M. (2009). "Long-time Preservation". Semantic Digital Libraries. S. Kruk and B. McDaniel. Springer-Verlag, Berlin, vol. XVI, pp 87.

RUBI (1999). "RUBI - Rede Universitária de Bibliotecas e Informação." <http://rubi.ua.pt/indexpt.html> [Janeiro de 2000]

Schoder, D., Fischbach, K. and Schmitt, C. (2005). "Core concepts in peer-to-peer (P2P) networking." P2P Computing: The Evolution of a Disruptive Technology. R. Subramanian and B. Goodman. Idea Group Inc, Hershey, PA.

SinBAD (2007). "SinBAD - Sistema Integrado para Biblioteca e Arquivo Digitais." <http://sinbad.ua.pt/> [Outubro de 2009]

Soergel, D. (2009). "Digital Libraries and Knowledge Organization". Semantic Digital Libraries. S. Kruk and B. McDaniel. Springer-Verlag, Berlin, vol. XVI, pp 9.

Stanford (2004). "Stanford Digital Library Technologies." <http://diglib.stanford.edu:8091/diglib/> [Setembro de 2009]

Staples, T., Wayland, R. and Payette, S. (2003). "The Fedora Project: An Open-source Digital Object Repository Management System." <http://www.dlib.org/dlib/april03/staples/04staples.html> [Abril de 2009]

Suleman, H. and Fox, E. A. (2001). "A Framework for Building Open Digital Libraries." D-Lib Magazine, vol. 7, no. 12. <http://www.dlib.org/dlib/december01/suleman/12suleman.html> [Julho de 2010]

Tarrant, D., O'Steen, B., Brody, T., Hitchcock, S., Jefferies, N. and Carr, L. (2009). "Using OAI-ORE to Transform Digital Repositories into Interoperable Storage and Services Applications." The Code4Lib Journal, no. 6. <http://journal.code4lib.org/articles/1062> [Julho de 2010]

Tian, M., Voigt, T., Naumowicz, T., Ritter, H. and Schiller, J. (2003). "Performance Impact of Web Services on Internet Servers". International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), Marina Del Rey, USA.

ULK (2006). "KVK – Karlsruhe Virtueller Katalog (University Library Karlsruhe)." <http://www.ubka.uni-karlsruhe.de/kvk.html> [Outubro de 2009]

UNESCO (2009). "UNESCO - United Nations Educational, Scientific and Cultural Organization." <http://portal.unesco.org> [Agosto de 2009]

Unicode (2009). "The Unicode Standard - Version 5.2.0." The Unicode Consortium. <http://www.unicode.org/versions/Unicode5.2.0/> [Dezembro de 2009]

Vaidya, P. and Plale, B. (2003). "Technical Report TR585: Benchmark Evaluation of XIndice as a Grid Information Server". Indiana University - School of Informatics and Computing. <http://www.cs.indiana.edu/pub/techreports/TR585.pdf> [Outubro de 2009]

Van de Sompel, H., Nelson, M., Lagoze, C. and Warner, S. (2004). "Resource Harvesting within the OAI-PMH Framework." D-Lib Magazine, vol. 10, no. 12. <http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html> [Julho de 2010]

W3C (2004a). "Architecture of the World Wide Web, Volume One." W3C. <http://www.w3.org/TR/webarch/> [Outubro de 2009]

W3C (2008). "Extensible Markup Language (XML) 1.0 (Fifth Edition)." W3C. <http://www.w3.org/TR/2008/REC-xml-20081126/> [Outubro de 2009]

W3C (1999b). "HTML 4.01 Specification." W3C. <http://www.w3.org/TR/html401/> [Agosto de 2009]

W3C (2004c). "Resource Description Framework (RDF): Concepts and Abstract Syntax." W3C. <http://www.w3.org/TR/rdf-concepts/> [Outubro de 2009]

W3C (2009). "Semantic Web." W3C. <http://www.w3.org/standards/semanticweb/> [Outubro de 2009]

W3C (2007a). "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)." W3C. <http://www.w3.org/TR/soap12-part1/> [1/12/2009]

W3C (2001). "URIs, URLs, and URNs: Clarifications and Recommendations 1.0." W3C. <http://www.w3.org/TR/uri-clarification/> [Outubro de 2009]

W3C (1999a). "Web Content Accessibility Guidelines 1.0." <http://www.w3.org/TR/WAI-WEBCONTENT/> [Outubro de 2009]

W3C (2004b). "Web Services Architecture." W3C. <http://www.w3.org/TR/ws-arch/> [Outubro de 2009]

W3C (2002). "XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)." W3C. <http://www.w3.org/TR/xhtml1/> [Novembro de 2009]

W3C (1999d). "XML Path Language (XPath) - Version 1.0." W3C. <http://www.w3.org/TR/xpath/> [Agosto de 2009]

W3C (2007b). "XQuery 1.0: An XML Query Language." W3C. <http://www.w3.org/TR/xquery/> [Dezembro de 2009]

W3C (1999c). "XSL Transformations (XSLT) - Version 1.0." W3C. <http://www.w3.org/TR/xslt> [Agosto de 2009]

Waters, D. J. (1998). "What are digital libraries?" CLIR (Council on Library and Information Resources), no. 4. <http://www.clir.org/pubs/issues/issues04.html> [Setembro de 2009]

Wells, A., Pearce, J., Groom, L. and Lee, B. (1998). "Connecting and Sharing: the Emerging Role of Z39.50 in Library Networks". VALA Conference, Melbourne.

Wells, H. G. (1937). "World Brain: The Idea of a Permanent World Encyclopaedia." <https://sherlock.ischool.berkeley.edu/wells/world_brain.html> [Dezembro de 2008]

Wells, H. G. (1938). "World Brain." Doubleday, Doran & Co., Inc., Garden City, N.Y.

Wilensky, R. (2000). "Digital library resources as a basis for collaborative work." Journal of the American Society for Information Science, vol. 51, no. 3, pp. 228-245

XmlDB (2003). "XML:DB." The XML:DB Initiative. <http://xmldb-org.sourceforge.net> [Dezembro de 2009]

XMLRPC (2009). "XML-RPC Specification." XML-RPC.COM. <http://www.xmlrpc.com/spec> [Dezembro de 2009]

Zagalo, H. T., Pinto, J. S. and Martins, J. A. (2000). "Sistema de Pesquisas Distribuídas e Paralelas em Sistemas Bibliográficos". 3ª Conferência sobre Redes de Computadores, CRC 2000, FCCN e Universidade de Aveiro, Viseu, Portugal.

Zagalo, H. T., Pinto, J. S. and Martins, J. A. (2001). "A Virtual Library Based on the Z39.50 Protocol". 3ª Conferência de Telecomunicações, ConfTele 2001, Instituto de Telecomunicações, Figueira da Foz, Portugal.

ZIG (1997a). "ATS-1 Profile." Library of Congress – Z39.50 International Standard Maintenance Agency. <http://www.loc.gov/z3950/agency/profiles/ats.html> [Abril de 2000]

ZIG (2003). "Information Retrieval (Z39.50): Application Service Definition and Protocol Specification." Library of Congress – Z39.50 International Standard Maintenance Agency. <http://www.loc.gov/z3950/agency/> [Janeiro de 2004]

ZIG (1997b). "ZDSR Profile - Z39.50 Profile for Simple Distributed Search and Ranked Retrieval." Library of Congress – Z39.50 International Standard Maintenance Agency. <http://www.loc.gov/z3950/agency/profiles/zdsr.html> [Abril de 2000]